

UNIVERSIDADE FEDERAL DO PARANÁ

WILLIAN ZALEWSKI

MODELAGEM SIMBÓLICA DE PADRÕES MORFOLÓGICOS  
PARA A CLASSIFICAÇÃO DE SÉRIES TEMPORAIS

CURITIBA

2015

WILLIAN ZALEWSKI

MODELAGEM SIMBÓLICA DE PADRÕES MORFOLÓGICOS  
PARA A CLASSIFICAÇÃO DE SÉRIES TEMPORAIS

Tese apresentada como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação, no Programa de Pós-graduação em Informática, Setor de Ciências Exatas da Universidade Federal do Paraná.  
Orientador: Dr. Fabiano Silva, Prof.

CURITIBA

2015

---

Z22m

Zalewski, Willian

Modelagem simbólica de padrões morfológicos para a classificação de séries temporais/ Willian Zalewski. – Curitiba, 2015.  
176 f. : il. color. ; 30 cm.

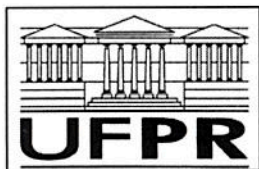
Tese - Universidade Federal do Paraná, Setor de Ciências Exatas,  
Programa de Pós-graduação em Informática, 2015.

Orientador: Fabiano Silva .  
Bibliografia: p. 149-167.

1. Mineração de dados (Computação). 2. Aprendizado do computador. 3.  
Análise de séries temporais. 4. Classificação (Computadores). 5. Sinais e  
símbolos. I. Universidade Federal do Paraná. II. Silva, Fabiano. III. Título.

CDD: 006.312

---



Ministério da Educação  
Universidade Federal do Paraná  
Programa de Pós-Graduação em Informática

### PARECER

Nós, abaixo assinados, membros da Comissão Examinadora da defesa do(a) aluno(a) de Doutorado em Ciência da Computação, Willian Zalewski, avaliamos a de tese de doutorado intitulado “Modelos Simbólicos de Padrões Morfológicos para a Classificação de Séries Temporais”, cuja defesa pública, foi realizada no dia 14 de setembro de 2015. Após avaliação, decidimos pela:

☒ **Aprovação** do(a) candidato(a). ( ) **Reprovação** do(a) candidato(a).

Curitiba, 14 de setembro de 2015.

Prof. Dr. Fabiano Silva  
PPGINF – Orientador

Prof. Dr. Gustavo Enrique de Almeida Prado Alves Batista  
ICMC/USP – Membro Externo

Prof. Dr. Fabrício Enembreck  
PUCPR – Membro Externo

AUSENTE

Prof. Dr. Luiz Eduardo Soares de Oliveira  
DINF/UFPR – Membro Externo

Prof. Dr. Eduardo Jaques Spinosa  
PPGINF – Membro Interno

Prof. Dr. Marcos Alexandre Castilho  
PPGINF – Membro Interno



*À José Ubirapena de Oliveira*

*À Sandra Zalewski*

*À Dabna Hellen Tomim*

## **AGRADECIMENTOS**

Ao professor Fabiano Silva pela dedicação com a qual conduziu a orientação deste trabalho, pela compreensão em todas as etapas do desenvolvimento, em especial no período em que estive em Pato Branco. Pessoa pela qual sou muito grato pela confiança depositada e pela oportunidade de ter compartilhado essa experiência.

À minha querida companheira de todas as jornadas, Dabna Hellen Tomim, pelo carinho e sempre apoio. Em especial pela compreensão e paciência durante todo o desenvolvimento deste trabalho. Também a seus pais José Carlos Tomim e Fátima Minto Tomim, pelo incentivo e confiança.

Aos meus pais, José Ubirapena de Oliveira e Sandra Zalewski, pela educação que me proporcionaram e pelo sempre incentivo às minhas escolhas em todas as etapas de minha vida.

Aos amigos André Gustavo Maletzke, Carlos Andres Ferrero, Everton Alvares Cherman e Joylan Nunes Maciel pelo apoio e auxílio em todos os aspectos que envolveram a realização deste trabalho.

Aos amigos Neimar Neitzel, Rafaella Aline Lopes da Silva Neitzel e Juliana Nadal, pelo importante apoio nessa jornada.

Ao Laboratório de Inteligência Artificial e Métodos Formais (LIAMF) por possibilitar o desenvolvimento deste trabalho. Em especial aos colegas Razer Anthom Nizer Rojas Montaña, Bruno César Ribas, Marcos Antonio Schreiner, Ricardo Tavares de Oliveira, Rodolfo Aparecido Rodovalho, Clariane Menezes e Eduardo Sant'Ana da Silva.

À UTFPR e aos colegas do DAINF-PB, pelo apoio para a conclusão deste trabalho. Em especial à Bruno César Ribas, um grande parceiro nessa empreitada, e à Richardson Ribeiro, Marco Antonio de Castro Barbosa e Marcelo Teixeira.

Ao C3SL/DINF pela infraestrutura computacional disponibilizada para a realização dos experimentos.

*“A felicidade, em suma, obtém-se por  
meio da vida contemplativa, uma vida intelectual  
sossegada, longe das perturbações do cotidiano.”  
(Aristóteles)*

## RESUMO

O contínuo armazenamento de dados ao longo do tempo, tais como séries temporais, tem motivado o desenvolvimento de novas abordagens baseadas em métodos de mineração de dados. Nesse cenário, uma nova área de pesquisa emergiu durante as últimas duas décadas, a mineração de dados em séries temporais. Mais especificamente, as abordagens baseadas em técnicas de aprendizado de máquina têm apresentado maior interesse entre os pesquisadores. Dentre as tarefas de mineração de dados, a classificação de séries temporais tem sido amplamente explorada, de modo que estudos recentes, utilizando algoritmos de aprendizado não simbólicos, têm reportado resultados significativos, em termos da acurácia de classificação. No entanto, em aplicações que envolvem processos de auxílio à tomada de decisão, tais como diagnóstico médico, controle de produção industrial, sistemas de monitoração de segurança em aeronaves ou usinas de energia elétrica, é necessário possibilitar o entendimento do raciocínio utilizado no processo de classificação. A primitiva *shapelet* foi proposta na literatura como um descritor de características morfológicas locais para possibilitar melhor compreensão dos conceitos, devido a sua maior proximidade com a percepção humana na identificação de padrões em séries temporais. Contudo, a maioria dos trabalhos relacionados ao estudo dessa primitiva tem se dedicado ao desenvolvimento de abordagens mais eficientes em termos de tempo e de acurácia, desconsiderando a necessidade da inteligibilidade dos classificadores. Nesse contexto, neste trabalho foi proposto um método que utiliza a transformada *shapelet* para a construção de modelos simbólicos de classificação por meio de uma abordagem híbrida que combina a representação de árvore de decisão com o algoritmo vizinho mais próximo. Também, foram desenvolvidas estratégias para melhorar a qualidade de representação da transformada *shapelet* na utilização de classificadores simbólicos, como árvores de decisão. Para avaliar o desempenho dessas propostas, foi conduzida uma avaliação experimental que envolveu a comparação com os algoritmos considerados estado da arte usando conjuntos de dados amplamente estudados na literatura de classificação de séries temporais. Com base nos resultados e análises realizadas nesta tese, foi possível verificar que a melhoria do processo de identificação de *shapelets* possibilita a construção de classificadores inteligíveis e competitivos; e que métodos híbridos podem contribuir para prover uma representação simbólica dos modelos, com desempenho equivalente ou até mesmo superior aos métodos não simbólicos.

**Palavras-chave:** mineração de dados. aprendizado de máquina. séries temporais. classificação. modelos simbólicos.



## ABSTRACT

The large amount of stored data over time, such as time series, has motivated the development of new approaches based on data mining methods. In this context, a new research area has emerged over the last two decades, the time series data mining. In particular, the approaches based on machine learning techniques have shown large interest among researchers. Among the data mining tasks, the time series classification has been widely exploited. Recent studies using non-symbolic learning algorithms have reported significant results in terms of classification accuracy. However, in applications related to decision making process, such as medical diagnosis, industrial production control, security monitoring systems in aircraft and in power plants, it is necessary allow the understanding of the reasoning used in the classification process. To take this into account, the shapelet primitive has been proposed in the literature as a descriptor of local morphological characteristics, which is closer to human perception for patterns identification in time series. On the other hand, most of the existing work related to shapelets has been dedicated to the development of more effective approaches in terms of time and accuracy, disregarding the need for interpretability of the classifiers. In this work, we propose to build symbolic models for time series classification using the shapelet transformation. This method is based on a hybrid approach that merges the decision tree representation and the nearest neighbor algorithm. Also, we developed strategies to improve the representation quality of the shapelet transformation using feature selection algorithms. We performed an experimental evaluation to analyze the performance of our proposals in comparison to the algorithms considered state of the art using datasets widely studied in the literature of time series classification. Based on the results and analysis carried out in this thesis, we found that the improvement of shapelet representation allows the construction of interpretable and competitive classifiers. Moreover, we found that the hybrid methods can help to provide symbolic models with equivalent or even superior performance to non-symbolic methods.

**Keywords:** data mining. machine learning. time series. classification. symbolic models.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Representação ilustrativa das componentes de tendência e sazonalidade. . . . .	35
Figura 2 – Representação gráfica da aplicação das técnicas de normalização (b) <i>offset</i> , (c) amplitude e (d) escala, sobre as séries temporais (a) <i>U</i> e <i>F</i> . . . . .	39
Figura 3 – Representação ilustrativa dos principais métodos da literatura de representação de séries temporais. . . . .	40
Figura 4 – Exemplo de uma matriz de distância obtida a partir da aplicação da DTW. . . . .	46
Figura 5 – Representação ilustrativa da diferença de alinhamento entre as distâncias Euclidiana e DTW. . . . .	47
Figura 6 – Ilustração da primitiva <i>shapelet</i> . . . . .	58
Figura 7 – Abordagens baseadas na primitiva <i>shapelet</i> . . . . .	59
Figura 8 – Ilustração do processo de aplicação de uma janela deslizante. . . .	61
Figura 9 – Representação esquemática da localização da menor distância encontrada entre as subsequências <i>S</i> e <i>R</i> . . . . .	64
Figura 10 – Representação esquemática de uma linha de ordenação. . . . .	65
Figura 11 – Ilustração do limiar de separação . . . . .	66
Figura 12 – Representação esquemática da construção de uma árvore de decisão construída com <i>shapelets</i> . . . . .	68
Figura 13 – Ilustração da ideia do algoritmo <i>logical shapelets</i> . . . . .	70
Figura 14 – Ilustração da desvantagem da abordagem exaustiva. . . . .	81
Figura 15 – Representação esquemática da configuração dos experimentos. . .	86
Figura 16 – Diagrama de diferença crítica para o pós-teste de Bonferroni-Dunn. .	90
Figura 17 – Comparação entre TSk e TS, em termos de acurácia no conjunto de teste. . . . .	91
Figura 18 – Comparação entre TS* e TS, em termos de acurácia no conjunto de teste. . . . .	92
Figura 19 – Comparação entre as variações da abordagem reduzida e TS, em termos de acurácia no conjunto de teste. . . . .	93
Figura 20 – Diagrama de diferença crítica para o pós-teste de Nemenyi. . . . .	96
Figura 21 – Comparação entre as variações da abordagem reduzida e os algoritmos da abordagem embutida (LS e FS), em termos de acurácia no conjunto de teste. . . . .	97

Figura 22 – Comparação entre as variações da abordagem reduzida e os algoritmos estado da arte (1NN ED e 1NN DTW), em termos de acurácia no conjunto de teste. . . . .	98
Figura 23 – Comparação entre as variações da abordagem reduzida e os algoritmos estado da arte (1NN ED e 1NN DTW), utilizando a análise <i>Texas Sharpshooter Fallacy</i> . . . . .	100
Figura 24 – Representação esquemática do método proposto. . . . .	105
Figura 25 – Representação gráfica da árvore de decisão. . . . .	110
Figura 26 – (a) Tabela-verdade e (b) representação gráfica da ADB para a função $f = (x_1 \vee x_2) \wedge (x_2 \vee x_3)$ . . . . .	111
Figura 27 – Representação gráfica do ROBDD da função $f = (x_1 \vee x_2) \wedge (x_2 \vee x_3)$ . . . . .	112
Figura 28 – (a) Tabela e (b) representação gráfica do respectivo ADD. . . . .	113
Figura 29 – (a) Tabela das funções $f$ e $g$ ; (b) representação gráfica dos ADDs $F$ e $G$ . . . . .	113
Figura 30 – Representação gráfica das operações de (a) disjunção, (b) conjunção e (c) máximo entre os ADDs $F$ e $G$ . . . . .	114
Figura 31 – Representação gráfica do (c) ADD resultante da operação máximo entre os ADDs (a) $A(c_1)$ e (b) $A(c_1)$ . . . . .	115
Figura 32 – Representação gráfica do (b) ADD resultante da operação máximo entre o ADD (a) $A(c_3)$ e o ADD $\max(A(c_1), A(c_2))$ da figura 31 (c). . . . .	116
Figura 33 – (a) Tabela parcialmente descrita e (b) representação gráfica do modelo parcial. . . . .	117
Figura 34 – Representação esquemática da configuração dos experimentos. . . . .	120
Figura 35 – Quantidade de atributos utilizada pelos algoritmos J48 e, HID e NNGE com abordagem <i>wrapper</i> e abordagem SSA, considerando cada uma das estratégias de binarização em cada conjunto de dados avaliado. . . . .	124
Figura 36 – Diagrama de diferença crítica para o pós-teste de Nemenyi, análise do algoritmo base J48. . . . .	129
Figura 37 – Diagrama de diferença crítica para o pós-teste de Nemenyi, análise do algoritmo base NNGE. . . . .	130
Figura 38 – Diagrama de diferença crítica para o pós-teste de Nemenyi, análise do algoritmo base HID. . . . .	130
Figura 39 – Diagrama de diferença crítica para o pós-teste de Nemenyi - Avaliação 3 . . . . .	131
Figura 40 – Comparação entre os algoritmos (a) HID e J48, (b) HID e NNGE, e (c) NNGE e J48, em termos de acurácia no conjunto de teste. . . . .	132
Figura 41 – Comparação entre os algoritmos (a) HID e J48, (b) HID e NNGE, e (c) NNGE e J48, utilizando a análise <i>Texas Sharpshooter Fallacy</i> . . . . .	133

Figura 42 – Diagrama de diferença crítica para o pós-teste de Bonferroni-Dunn - Avaliação 4. . . . .	136
Figura 43 – Comparação entre o algoritmo HID e os algoritmos da abordagem embutida (LS e FS), da abordagem transformada (TS+CS, TS+CFS e TS+FCBF) e do estado da arte (1NN ED e 1NN DTW), em termos de acurácia no conjunto de teste. . . . .	137
Figura 44 – Comparação entre o algoritmo HID e os algoritmos da abordagem embutida (LS e FS), da abordagem transformada (TS+CS, TS+CFS e TS+FCBF) e do estado da arte (1NN ED e 1NN DTW), pela análise <i>Texas Sharpshooter Fallacy</i> . . . . .	138
Figura 45 – Representação da árvore de decisão gerada pelo algoritmo HID para o conjunto dados <i>CBF</i> . . . . .	140
Figura 46 – Representação da árvore de decisão gerada para o conjunto dados <i>FaceFour</i> . . . . .	140
Figura 47 – Representação da árvore de decisão gerada para o conjunto dados <i>Beef</i> . . . . .	141

## LISTA DE TABELAS

Tabela 1 – Representação atributo-valor da transformada <i>shapelet</i> . . . . .	74
Tabela 2 – Descrição dos conjuntos de dados utilizados. . . . .	85
Tabela 3 – Valores de $q_{0,05}$ para diferentes valores de $K$ no pós-teste de Nemenyi e de Bonferroni-Dunn. . . . .	89
Tabela 4 – Resultados dos experimentos realizados para a Avaliação 1. . . . .	89
Tabela 5 – Resultados das variações da abordagem reduzida, em termos de vitórias, derrotas e empates, em relação à abordagem TS. . . . .	93
Tabela 6 – Resultados dos experimentos realizados para a Avaliação 2. . . . .	95
Tabela 7 – Resultados das variações da abordagem reduzida, em termos de vitórias, derrotas e empates, em relação a abordagem embutida (LS e FS). . . . .	97
Tabela 8 – Resultados das variações da abordagem reduzida, em termos de vitórias, derrotas e empates, em relação aos algoritmos estado da arte (1NN ED e 1NN DTW). . . . .	99
Tabela 9 – Sumário dos resultados da análise <i>Texas Sharpshooter Fallacy</i> . . .	101
Tabela 10 – Descrição do conjunto adicional de dados utilizado. . . . .	119
Tabela 11 – Resultados da comparação entre as abordagens SSA e <i>wrapper</i> para o algoritmo HID. . . . .	123
Tabela 12 – Resultados da comparação entre as abordagens SSA e <i>wrapper</i> para o algoritmo NNGE. . . . .	123
Tabela 13 – Resultados dos experimentos realizados utilizando o J48 como algoritmo base, . . . . .	126
Tabela 14 – Resultados dos experimentos realizados utilizando o NNGE como algoritmo base. . . . .	127
Tabela 15 – Resultados dos experimentos realizados utilizando o HID como algoritmo base. . . . .	128
Tabela 16 – Sumário dos resultados da análise <i>Texas Sharpshooter Fallacy</i> . . .	133
Tabela 17 – Resultados dos experimentos realizados utilizando o HID em comparação com os algoritmos LS, FS, 1NN-ED, 1NN-DTW, TS+CFS, TS+CS e TS+FCBF. . . . .	135
Tabela 18 – Resultados da comparação entre o algoritmo HID e os demais algoritmos analisados, em termos de vitórias, derrotas e empates. . . .	136
Tabela 19 – Sumário dos resultados da análise <i>Texas Sharpshooter Fallacy</i> . . .	138
Tabela A.1 – Quantidade de atributos utilizada, considerando a estratégia de aprendizado de limiares, ganho de informação (lg). . . . .	171

Tabela A.2—Quantidade de atributos utilizada, considerando a estratégia de aprendizado de limiares, agrupamento (2k). . . . .	172
Tabela A.3—Quantidade de atributos utilizada, considerando a estratégia de aprendizado de limiares, precisão (Pr). . . . .	173
Tabela B.1—Resultados dos experimentos realizados utilizando o HID como algoritmo base, sem seleção de atributos (SSA). . . . .	175
Tabela B.2—Resultados dos experimentos realizados utilizando o NNGE como algoritmo base, sem seleção de atributos (SSA). . . . .	176

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>25</b>
1.1 Objetivos	27
1.2 Principais Contribuições	29
1.3 Organização do trabalho	29
<b>2 MINERAÇÃO DE DADOS EM SÉRIES TEMPORAIS</b>	<b>31</b>
2.1 Considerações iniciais	31
2.2 Mineração de Dados	31
2.3 Aprendizado de Máquina	32
2.4 Fundamentos de séries temporais	34
2.5 Análise de séries temporais	36
2.6 Tarefas de pré-processamento	37
2.6.1 Amostragem, tendência e ruído	38
2.6.2 Normalização	38
2.6.3 Representação	40
2.7 Medidas de Similaridade	44
2.7.1 Medidas baseadas na forma	45
2.7.2 Medidas baseadas em características	47
2.7.3 Medidas baseadas na edição	48
2.7.4 Medidas baseadas na estrutura	48
2.8 Aprendizado de máquina em séries temporais	48
2.8.1 Recuperação por conteúdo	49
2.8.2 Agrupamento	49
2.8.3 Descoberta de regras	49
2.8.4 Previsão de valores	50
2.8.5 Identificação de eventos	50
2.8.6 Classificação de séries temporais	51
2.9 Considerações Finais	54
<b>3 INTELIGIBILIDADE NA CLASSIFICAÇÃO DE SÉRIES TEMPORAIS</b>	<b>55</b>
3.1 Considerações iniciais	55
3.2 Trabalhos relacionados	55
3.3 Primitiva <i>shapelet</i>	58
3.4 Abordagem embutida – Algoritmo YK	59
3.4.1 Gerar subsequências candidatas	61
3.4.2 Calcular distâncias	63

3.4.3 Determinar qualidade . . . . .	64
3.4.4 Construção da árvore de decisão . . . . .	67
3.4.5 Classificação . . . . .	69
3.4.6 Escalabilidade . . . . .	69
3.4.7 <i>Logical shapelets</i> . . . . .	70
3.4.8 <i>Fast shapelets</i> . . . . .	71
3.5 Abordagem transformada . . . . .	71
3.5.1 Estimar o parâmetro $k$ . . . . .	72
3.5.2 Selecionar $k$ <i>shapelets</i> . . . . .	72
3.5.3 Aplicar a transformada . . . . .	74
3.5.4 <i>Clustering shapelets</i> . . . . .	75
3.5.5 <i>Discriminative shapelets</i> . . . . .	76
3.6 Generalizações e aplicações de <i>shapelets</i> . . . . .	76
3.7 Considerações finais . . . . .	77
<b>4 EXPLORANDO A TRANSFORMADA SHAPELET . . . . .</b>	<b>79</b>
4.1 Considerações iniciais . . . . .	79
4.2 Extensões da transformada <i>shapelet</i> . . . . .	79
4.2.1 Abordagem exaustiva – TSk . . . . .	80
4.2.2 Abordagem relaxada – TS* . . . . .	80
4.2.3 Abordagem reduzida – TS+ . . . . .	81
4.3 Avaliação experimental . . . . .	84
4.3.1 Algoritmos utilizados . . . . .	84
4.3.2 Descrição dos conjuntos de dados . . . . .	85
4.3.3 Configuração dos experimentos . . . . .	86
4.3.4 Análise de significância estatística . . . . .	87
4.3.5 Avaliação 1 - Resultados e Discussão . . . . .	89
4.3.6 Avaliação 2 - Resultados e Discussão . . . . .	94
4.4 Considerações finais . . . . .	101
<b>5 ABORDAGEM HÍBRIDA EM DIAGRAMAS DE DECISÃO . . . . .</b>	<b>103</b>
5.1 Considerações iniciais . . . . .	103
5.2 Trabalhos relacionados . . . . .	103
5.3 Método proposto . . . . .	104
5.4 Representação binária . . . . .	106
5.4.1 Aprendizado de limiares . . . . .	106
5.4.2 Transformação binária . . . . .	108
5.4.3 Representação binária reduzida . . . . .	109
5.5 Representação simbólica dos modelos . . . . .	110
5.5.1 Diagramas de decisão . . . . .	110



5.5.2 HID: <i>Hybrid Instance based Decision tree</i> . . . . .	113
5.5.3 Classificação . . . . .	118
5.6 Avaliação experimental . . . . .	118
5.6.1 Descrição dos conjuntos de dados . . . . .	118
5.6.2 Algoritmos utilizados . . . . .	119
5.6.3 Configuração dos experimentos . . . . .	120
5.6.4 Avaliação 1 – Resultados e discussão . . . . .	122
5.6.5 Avaliação 2 – Resultados e discussão . . . . .	125
5.6.6 Avaliação 3 – Resultados e discussão . . . . .	131
5.6.7 Avaliação 4 – Resultados e discussão . . . . .	134
5.7 Representação Simbólica HID . . . . .	139
5.8 Considerações finais . . . . .	141
<b>6 CONCLUSÃO . . . . .</b>	<b>143</b>
6.1 Explorando a transforma <i>shapelet</i> - capítulo 4 . . . . .	143
6.2 Abordagem híbrida em diagramas de decisão - capítulo 5 . . . . .	144
6.3 Limitações . . . . .	146
6.4 Trabalhos futuros . . . . .	147
<b>REFERÊNCIAS . . . . .</b>	<b>149</b>
 <b>APÊNDICE A QUANTIDADE DE ATRIBUTOS (<i>SHAPELETS</i>) . . . . .</b>	 <b>171</b>
<b>APÊNDICE B RESULTADOS DE ACURÁCIA PARA SSA (HID E NNGE) . . . . .</b>	<b>175</b>

## 1 INTRODUÇÃO

O desenvolvimento tecnológico referente à capacidade de armazenamento e ao processamento de dados, aliado à redução de custos desses recursos, tem facilitado a aquisição de dados ao longo do tempo e possibilitado o surgimento de bases de dados com grande volume de informação sequencial. Nesse cenário, quase todo tipo de dado coletado a partir de atividades humanas, fenômenos naturais ou processos biológicos, pode estar sujeito à variação temporal. O tipo de dado temporal mais comum é a série temporal, a qual pode ser entendida como um conjunto ordenado de observações registradas cronologicamente ([MORETTIN; TOLOI, 2006](#)). Séries temporais estão presentes em diversas áreas do conhecimento, como exemplo, economia (preços diários de ações, taxa mensal de desemprego, produção industrial), medicina (eletrocardiograma, eletroencefalograma), epidemiologia (número mensal de novos casos de meningite), meteorologia (precipitação pluviométrica, temperatura diária, velocidade do vento) ([EHLERS, 2005](#)).

A utilização de métodos de análise para o estudo de eventos e comportamentos contidos nas séries temporais consiste em uma tarefa não trivial e dependente do domínio de aplicação ([FERRERO, 2009](#); [MALETZKE et al., 2014](#)). Nesse contexto, abordagens baseadas em métodos de mineração de dados têm sido amplamente exploradas para a análise de séries temporais nas últimas duas décadas ([ANTUNES; OLIVEIRA, 2001](#); [LAST; KLEIN; KANDEL, 2001](#); [LAST; KANDEL; BUNKE, 2004](#); [LAXMAN; SASTRY, 2006](#); [MORCHEN, 2006](#); [FU, 2011](#); [ESLING; AGON, 2012](#)). Dentre as tarefas de mineração de dados, a classificação de séries temporais tem sido amplamente explorada. No entanto, os algoritmos tradicionais de classificação foram desenvolvidos para tratar dados sem considerar a existência de relações de ordem ou de tempo ([BAGNALL et al., 2012](#)). Desse modo, uma grande quantidade de estudos, com diferentes tipos de abordagens para considerar a dependência temporal dos dados, tem sido propostos na literatura nos últimos anos ([KEOGH; PAZZANI, 1998](#); [GEURTS, 2001](#); [NANOPOULOS; ALCOCK; MANOLOPOULOS, 2001](#); [KADOUS; SAMMUT, 2004](#); [COTOFREI; STOFFEL, 2002](#); [XING; PEI; KEOGH, 2010](#); [BAGNALL et al., 2012](#); [SOUZA; SILVA; BATISTA, 2014](#)).

Trabalhos recentes ([DING et al., 2008](#); [WANG et al., 2013](#)) verificaram por meio de uma extensa avaliação empírica, que o algoritmo de classificação 1-vizinho mais próximo (1NN) em combinação com a medida de distância *dynamic time warping* (DTW) apresenta o melhor desempenho, em termos de acurácia, para a maioria dos domínios de séries temporais avaliados, e tem sido apontado como o método estado-da-arte ([BATISTA; WANG; KEOGH, 2011](#)). No entanto, em aplicações que envolvem

processos de auxílio à tomada de decisão, tais como diagnóstico médico, controle de produção industrial, sistemas de monitoração de segurança em aeronaves ou usinas de energia elétrica, é necessário possibilitar o entendimento do raciocínio utilizado pelo processo de classificação (GEURTS, 2001; YAMADA et al., 2005; YE; KEOGH, 2009; MALETZKE, 2009; XING et al., 2011; XING; PEI; YU, 2012; GHALWASH; OBRADOVIC, 2012; GHALWASH; RADOSAVLJEVIC; OBRADOVIC, 2013). Essa característica é pouco presente no algoritmo 1NN-DTW, pois a única informação provida refere-se ao grau de similaridade entre as séries temporais consideradas semelhantes. Além desse aspecto, por se basear em uma estratégia de aprendizado *lazy*, nenhum modelo explícito sobre o conhecimento é produzido. Assim, a estratégia mais comum para a construção de classificadores inteligíveis tem sido a utilização de algoritmos de aprendizado de máquina simbólico, tais como árvores ou regras de decisão. Em geral, os métodos baseados em regras possibilitam maior facilidade de compreensão do conhecimento representado nos modelos (MICHALSKI; BRATKO; KUBAT, 1998; REZENDE, 2003; FACELI et al., 2011).

A maioria dos métodos propostos na literatura para a construção de classificadores simbólicos de séries temporais baseia-se em uma representação de características estatísticas e/ou de parâmetros de ajuste de funções (BAKSHI et al., 1994; KADOUS, 1999; KUDO; TOYAMA; SHIMBO, 1999; RODRIGUEZ; ALONSO; BOSTROM, 2001; MORCHEN, 2003; KADOUS; SAMMUT, 2004; KADOUS; SAMMUT, 2005; COTOFREI; STOFFEL, 2002; COTOFREI; STOFFEL, 2005; HIDASI; GASPAPANEK, 2011). Contudo, a utilização dessas representações pode prejudicar a compreensão do conhecimento mesmo em modelos simbólicos, pois referem-se a conceitos menos próximos da intuição humana. Para contornar esse problema, a primitiva *shapelet* (YE; KEOGH, 2009) foi proposta como um descritor de características morfológicas locais, que possibilita melhor compreensão dos conceitos, devido a sua maior proximidade com a percepção humana para a identificação de padrões em séries temporais (YE; KEOGH, 2011).

Desde sua introdução, a primitiva *shapelet* tem sido utilizada em diversas abordagens de classificação (MUEEN et al., 2009; XING et al., 2011; XING; PEI; YU, 2012; GHALWASH; RADOSAVLJEVIC; OBRADOVIC, 2013; YUAN; WANG; HAN, 2014). Em especial, no trabalho de (LINES et al., 2012) os autores propuseram a separação da indução do modelo de classificação do processo de identificação de *shapelets*, com o intuito de flexibilizar a adaptação de novas abordagens. Para isso, foi proposta a transformada *shapelet*, a qual consiste em representar cada série temporal por meio de um conjunto de valores de distâncias em relação a cada *shapelet* identificada. No entanto, a maior parte dos trabalhos relacionados ao estudo dessa primitiva tem se dedicado ao desenvolvimento de abordagens mais eficientes em termos de tempo e de acurácia (RAKTHANMANON; KEOGH, 2013; HE et al., 2012; GORDON; HENDLER;

ROKACH, 2012; HILLS et al., 2013; GRABOCKA et al., 2014, 2014; GRABOCKA; WISTUBA; SCHMIDT-THIEME, 2015). Esse cenário, apesar de positivo, demonstra também que o propósito original da primitiva *shapelet*, que refere-se a possibilitar maior grau de inteligibilidade à tarefa de classificação de séries temporais, ainda tem sido pouco explorado. Neste trabalho, investigamos como a transformada *shapelet* pode ser melhor utilizada para a representação de séries temporais, bem como para a classificação por meio de árvores de decisão.

Em estudos recentes (HILLS et al., 2013), a utilização da transformada *shapelet* por meio de classificadores não simbólicos tem apresentado melhores resultados do que com árvores de decisão, em termos de acurácia. Entretanto, como mencionado, as abordagens não simbólicas apresentam maior dificuldade para a interpretação do conhecimento (FACELI et al., 2011). Nesse contexto, outros estudos na literatura exploraram alternativas para permitir a combinação de estratégias simbólicas e não simbólicas. Em particular, algumas propostas basearam-se na combinação de abordagens que possibilitam a construção de regras de decisão com algoritmos baseados em exemplos, tais como o vizinho mais próximo (DOMINGOS, 1995; MARTIN, 1995; FRIEDMAN, 1996; FERN; BRODLEY, 2003; GAO; ESTER, 2006). A ideia desse tipo de abordagem consiste em aproveitar as vantagens enquanto minimiza-se as desvantagens de cada estratégia. Esse tipo de sistema híbrido de aprendizado é também objeto de estudo desta tese.

Dado esse contexto, um dos desafios para a tarefa de classificação de séries temporais refere-se ao desenvolvimento de modelos simbólicos que sejam competitivos em termos de acurácia.

## 1.1 OBJETIVOS

Considerando o cenário descrito sobre a tarefa de classificação de séries temporais, os objetivos desta tese são descritos a seguir:

### Objetivo geral

Melhorar a acurácia de classificação de séries temporais por meio de modelos simbólicos e padrões morfológicos.

### Objetivos específicos:

1. Propor estratégias para melhorar a qualidade da representação de séries temporais por meio da transformada *shapelet*, considerando a construção de modelos de classificação simbólicos baseados em árvores de decisão; e

2. Propor um método que permita utilizar a transformada *shapelet* para a construção de modelos simbólicos de classificação por meio de uma abordagem híbrida que combine a representação de árvore de decisão com o algoritmo vizinho mais próximo.

definidos sob as seguintes hipóteses:

1. A qualidade da representação da transformada *shapelet* tem influência no desempenho de classificadores simbólicos como árvores de decisão; e
2. A utilização da transformada *shapelet* por meio da combinação de algoritmos de aprendizado simbólico e baseados em exemplos possibilita a construção de modelos de classificação de séries temporais inteligíveis e de melhor desempenho em termos de acurácia.

baseadas nas seguintes assertivas:

- Algoritmos baseados no aprendizado simbólico, tais como árvores de decisão, permitem a construção de estruturas baseadas em regras, as quais podem facilitar a compreensão do conhecimento por seres humanos (MICHALSKI; BRATKO; KUBAT, 1998; REZENDE, 2003; FACELI et al., 2011);
- A análise de séries temporais por meio de características morfológicas que permitam descrever eventos e comportamentos apresenta maior proximidade com a percepção humana (GEURTS, 2001; YAMADA et al., 2003; BATYRSHIN; SHEREMETOV, 2008; YE; KEOGH, 2009; MALETZKE et al., 2009);
- Em problemas que envolvem a análise de características morfológicas, a transformação das séries temporais em um domínio alternativo de representação pode prover melhor acurácia do que a utilização de outras abordagens sobre o domínio do tempo (BAGNALL et al., 2012);
- Na tarefa de agrupamento de séries temporais existe a necessidade de ignorar determinadas subsequências (ZAKARIA; MUEEN; KEOGH, 2012; RAKTHANMANON et al., 2011) de modo que essa constatação é também válida para a tarefa de classificação de séries temporais;
- A identificação de *shapelets* permite a construção de modelos de classificação inteligíveis, com base em padrões morfológicos locais, em que as subsequências pouco discriminantes de um dado conjunto podem ser desconsideradas (YE; KEOGH, 2009; YE; KEOGH, 2011; RAKTHANMANON; KEOGH, 2013);

- A existência e a combinação de múltiplos *shapelets* é relevante para a classificação de séries temporais (MUEEN; KEOGH; YOUNG, 2011; HILLS et al., 2013);
- A separação do processo de identificação de *shapelets* da tarefa de indução da árvore de decisão flexibiliza a adaptação de novas abordagens de mineração de séries temporais (LINES et al., 2012; HILLS et al., 2013);
- A combinação de algoritmos de aprendizado simbólico e baseado em exemplos possibilita agregar as vantagens e minimizar as desvantagens de cada abordagem (SALZBERG, 1991; MARTIN, 1995; FRIEDMAN, 1996; FERN; BRODLEY, 2003; GAO; ESTER, 2006).

## 1.2 PRINCIPAIS CONTRIBUIÇÕES

A seguir são apresentadas as principais contribuições deste trabalho:

- Um estudo sobre a construção de modelos inteligíveis para a classificação de séries temporais, em especial aos relacionados à utilização da primitiva *shapelet*;
- Proposta de três estratégias para a melhoria da representatividade da transformada *shapelet*, sendo duas relacionadas à exploração de parâmetros mais adequados do algoritmo original, e uma que baseia-se na utilização de técnicas tradicionais para a seleção de subconjuntos de atributos;
- Método simbólico baseado em padrões morfológicos para a classificação de séries temporais, em que são propostos dois algoritmos:
  1. Algoritmo para a representação binária da transformada *shapelet* (*binary shapelet transform* – BST), para o qual foram propostas três abordagens de discretização;
  2. Algoritmo híbrido para a construção de modelos simbólicos de classificação de séries temporais (*hybrid instance based decision tree* – HID), que combina a estratégia de representação por meio de árvores de decisão com algoritmos baseados em exemplos.

## 1.3 ORGANIZAÇÃO DO TRABALHO

Os capítulos 2 e 3 estão relacionados à revisão bibliográfica utilizada como base para as contribuições propostas nos capítulos 4 e 5. Mais especificamente, esta tese está organizada do seguinte modo:

**Capítulo 2 – Mineração de dados em séries temporais:** nesse capítulo são apresentados os principais aspectos referentes à mineração de dados em séries temporais por meio da caracterização de tarefas como pré-processamento, medidas de similaridade, e abordagens baseadas em técnicas de aprendizado de máquina, com ênfase nos principais métodos de classificação de séries temporais propostos na literatura.

**Capítulo 3 – Inteligibilidade na classificação de séries temporais:** nesse capítulo são apresentados os principais estudos da literatura que se dedicaram à construção de classificadores inteligíveis para séries temporais. Em especial, são descritos os métodos baseados na primitiva *shapelet*, a qual tem apresentado resultados promissores como um descritor de características morfológicas interpretáveis.

**Capítulo 4 – Explorando a transformada *shapelet*:** nesse capítulo são apresentados os estudos conduzidos neste trabalho, que tiveram o intuito de aprimorar a qualidade da representação de séries temporais por meio da transformada *shapelet* para a construção de árvores de decisão. Inicialmente, são apresentadas três propostas de abordagens baseadas na transformada *shapelet*, com o objetivo de contornar algumas das desvantagens do algoritmo original. Posteriormente, é descrita a análise experimental realizada para avaliar essas propostas em relação aos algoritmos baseados na abordagem embutida, e em relação aos algoritmos considerados estado-da-arte para a classificação de séries temporais.

**Capítulo 5 – Abordagem híbrida em diagramas de decisão:** nesse capítulo é descrito o método proposto para a classificação de séries temporais por meio da construção de modelos simbólicos híbridos. Nesse contexto, é introduzido o algoritmo HID, o qual utiliza diagramas algébricos de decisão por meio da representação binária de descritores morfológicos, tais como *shapelets*. Posteriormente é descrita a avaliação experimental realizada com conjuntos de dados amplamente estudados na literatura de séries temporais, a qual envolveu também a comparação com algoritmos baseados na construção de modelos simbólicos e híbridos, bem como métodos considerados estado da arte para a classificação de séries temporais.

**Capítulo 6 – Conclusão:** nesse capítulo são apresentadas as conclusões deste trabalho, as principais contribuições, as limitações e os trabalhos futuros.



## 2 MINERAÇÃO DE DADOS EM SÉRIES TEMPORAIS

### 2.1 CONSIDERAÇÕES INICIAIS

A análise de comportamentos temporais é uma tarefa de crescente interesse em diversas áreas do conhecimento, tais como economia (preços diários de ações, taxa mensal de desemprego, produção industrial), medicina (eletrocardiograma, eletroencefalograma), epidemiologia (número mensal de novos casos de meningite), meteorologia (precipitação pluviométrica, temperatura diária, velocidade do vento) (EHLERS, 2005). A contínua coleta de informações ao longo do tempo tem contribuído para o surgimento de bases de dados com grandes volumes de informação sequencial, o que torna difícil a sua interpretação por seres humanos. Quando dados são coletados ao longo do tempo, estes podem ser representados por meio de uma série temporal (MORETTIN; TOLOI, 2006). A utilização de métodos de análise para o estudo de eventos contidos nas séries temporais consiste em uma tarefa não trivial e dependente do domínio de aplicação (FERRERO, 2009; MALETZKE et al., 2014). Nesse cenário, abordagens baseadas em métodos de mineração de dados têm sido amplamente exploradas para a análise de séries temporais (MORCHEN, 2006; FU, 2011; ESLING; AGON, 2012).

Neste capítulo são apresentados os principais conceitos e tarefas referentes à mineração de dados em séries temporais, em especial são relacionados os principais métodos de pré-processamento, medidas de similaridade e de aprendizado de máquina.

### 2.2 MINERAÇÃO DE DADOS

Nos últimos anos, o desenvolvimento tecnológico tem possibilitado o armazenamento de um grande volume de dados nos mais variados domínios do conhecimento. Como consequência do rápido acúmulo e da diversidade dos dados, a utilização de processos de análise manuais torna-se inviável, em virtude da quantidade de relações e associações possíveis. Outro aspecto originado a partir desse cenário refere-se ao armazenamento desestruturado dos dados, os quais são frequentemente registrados sob diferentes formatos, de modo que a análise direta sobre esses dados em sua forma original, é dificultada ou impossibilitada. Nesse contexto, o desenvolvimento de métodos e processos, por meio de técnicas computacionais, tem sido proposto na literatura para auxiliar no processamento automático ou semiautomático de grandes conjuntos de dados. Em especial, o processo de mineração de dados (MD) (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996; MITCHELL, 1997; MICHALSKI; BRATKO; KUBAT, 1998; WEISS; INDURKHYA, 1998; HAN; KAMBER, 2006) tem atraído a atenção de pesquisadores de diversas áreas.



O processo de MD tem como objetivo a extração de conhecimento sobre conjuntos de dados de um determinado domínio, que possa ser útil para especialistas em processos de tomada de decisão. Nesse cenário, o conhecimento pode ser caracterizado como a capacidade de relacionar informações por meio de modelos, que permitam descrever ou indicar as ações a serem realizadas (REZENDE, 2003). Portanto, um importante requisito refere-se à possibilitar que o conhecimento descoberto seja de fácil compreensão a humanos. O processo de MD pode ser estruturado em três principais etapas: pré-processamento, extração de padrões e pós-processamento, as quais podem estar relacionadas de modo interativo e iterativo (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996; WEISS; INDURKHYA, 1998).

**Pré-processamento:** essa etapa consiste na atividade de conhecer o domínio de aplicação e os tipos de dados a serem analisados. Também está relacionada com a preparação dos dados para um formato que seja adequado para a etapa de extração de padrões (PYLE, 1999). Algumas tarefas comumente aplicadas nessa etapa são (FACELI et al., 2011): integração de dados, transformação de dados, limpeza de dados, redução de dados (seleção de atributos e redução de exemplos).

**Extração de padrões:** essa etapa tem como objetivo a busca por padrões que permitam representar o conhecimento que possa existir, de modo implícito, no conjunto de dados analisado (WITTEN; FRANK, 2005). Nessa etapa, em geral, podem ser adotadas técnicas de diferentes áreas como visualização, estatística e inteligência artificial.

**Pós-processamento:** as atividades relacionadas com essa etapa correspondem à avaliação e à validação dos modelos produzidos na etapa anterior. Essas ações podem ser realizadas por meio da aplicação de medidas de desempenho e de qualidade, de testes estatísticos e/ou da verificação com especialistas da área.

## 2.3 APRENDIZADO DE MÁQUINA

Como mencionado, em virtude da quantidade crescente de dados armazenados e conseqüentemente do número de correlações necessárias no processo de análise, tornou-se cada vez mais importante o desenvolvimento de técnicas computacionais que pudessem induzir hipóteses de modo automático (ALPAYDIN, 2004), com o intuito de reduzir a intervenção humana e a dependência por especialistas do domínio. Nesse cenário, dentre as áreas comumente empregadas no processo de MD para a etapa de extração de padrões, a de inteligência artificial tem recebido a atenção de pesquisadores de diversas áreas (FACELI et al., 2011), em particular, pela utilização de técnicas de aprendizado de máquina (AM).

O principal meio para a aquisição de conhecimento, de modo automático, por meio dos algoritmos de AM é baseado na inferência indutiva, na qual novos conhecimentos podem ser derivados a partir de outros previamente conhecidos. Assim, os algoritmos de AM tem por objetivo a indução de uma hipótese a partir de um conjunto de exemplos que permitam caracterizar o domínio ou problema que se deseja tratar (MITCHELL, 1997).

No contexto de AM, de acordo com o tipo de tarefa a ser realizada, o aprendizado indutivo pode ser organizado em aprendizado supervisionado e aprendizado não supervisionado. No aprendizado supervisionado, os algoritmos de indução são aplicados sobre um conjunto de exemplos do domínio, chamado conjunto de treinamento, no qual o conceito que caracteriza cada exemplo (classe) é conhecido. Caso o domínio desse conceito seja composto por um conjunto de valores nominais, o algoritmo de indução produz um classificador. Caso o domínio seja um conjunto infinito e ordenado de valores, o algoritmo de indução gera um regressor. Um classificador (ou regressor) pode ser entendido como uma função, que dado um exemplo sem classe associada, atribui a esse exemplo uma das possíveis classes conhecidas (REZENDE, 2003). Em relação ao aprendizado não supervisionado, a classe dos exemplos contidos no conjunto de treinamento não é conhecida. Nesse tipo de aprendizado, as tarefas comumente abordadas são: sumarização, a qual tem por objetivo construir uma descrição representativa e compacta dos dados; associação, que consiste na identificação de padrões frequentes que estabeleçam algum tipo de relação entre os atributos; e agrupamento, no qual o objetivo é encontrar grupos de exemplos segundo alguma métrica de similaridade (FACELI et al., 2011).

Considerando os diferentes tipos de tarefas, domínios e objetivos relacionados à utilização dos algoritmos de AM, existem distintos critérios de organização que podem ser utilizados para auxiliar na escolha do algoritmo mais adequado para cada situação. Assim, de acordo tipo de conceito utilizado para induzir uma determinada hipótese, os algoritmos de aprendizado podem ser estruturados em paradigmas, os quais alguns são brevemente apresentados a seguir (REZENDE, 2003):

- **Paradigma simbólico:** os algoritmos baseados nesse paradigma realizam o processo de aprendizagem de um determinado conceito utilizando representações simbólicas por meio da análise de exemplos e contra-exemplos. Essas representações estão, geralmente, representadas na forma de expressão lógica, árvore de decisão ou rede semântica. Um exemplo de algoritmo desse paradigma é o C4.5 (QUINLAN, 1993), que baseia-se na indução de árvores de decisão;
- **Paradigma baseado em exemplos:** os algoritmos baseados nesse paradigma armazenam os exemplos e utilizam medidas de similaridade para identificar os

casos mais similares ao exemplo a ser analisado. Um exemplo de algoritmo desse paradigma é o k-vizinhos mais próximos (kNN) (AHA; KIBLER; ALBERT, 1991);

- **Paradigma estatístico:** os algoritmos baseados nesse paradigma utilizam modelos estatísticos para encontrar uma aproximação do conceito induzido. Dentre as abordagens existentes, destacam-se o aprendizado Bayesiano (MITCHELL, 1997) e máquinas de vetores de suporte (SVM) (CORTES; VAPNIK, 1995);
- **Paradigma conexionista:** os algoritmos desse paradigma baseiam-se em construções matemáticas inspiradas em conexões neuronais do sistema nervoso humano. As redes neurais artificiais (RNA) (HAYKIN, 1999) são exemplos de técnicas baseadas nesse paradigma;

Outro tipo de organização dos algoritmos de AM está relacionada ao grau de inteligibilidade. Os diferentes algoritmos de aprendizado existentes utilizam distintos modos de representação para descrever uma hipótese. Assim, os algoritmos de AM também podem ser classificados em duas categorias: sistemas caixa preta e orientados a conhecimento (MICHALSKI; BRATKO; KUBAT, 1998). A primeira caracteriza os algoritmos que baseiam-se em suas próprias representações dos conceitos, de modo que estas podem não ser facilmente interpretáveis por humanos. A segunda possibilita a construção de representações simbólicas que sejam mais próximas ao raciocínio humano. Além desse aspecto, os algoritmos de AM podem ser caracterizados como sistemas de aprendizado *eager* ou *lazy*. No aprendizado *eager*, os algoritmos utilizam um conjunto de exemplos para a induzir um modelo (hipótese). Enquanto que no aprendizado *lazy* não é realizada a construção de um modelo explícito, e portanto o conjunto de exemplos de treinamento deve ser mantido durante o processo de análise de novos exemplos.

Como será abordado nas próximas seções, os algoritmos tradicionais de aprendizado de máquina foram desenvolvidos para o tratamento de dados que não apresentam correlação entre os atributos que descrevem os exemplos, o que é uma das principais características de alguns tipos de dados, como as séries temporais. Nesse contexto, serão descritas as principais tarefas de MD e técnicas de AM desenvolvidas para a análise de séries temporais.

## 2.4 FUNDAMENTOS DE SÉRIES TEMPORAIS

Uma série temporal consiste em um conjunto de observações de um determinado fenômeno realizadas de modo sequencial ao longo do tempo, não necessariamente igualmente espaçadas. Esse tipo de dado tem como característica fundamental a dependência entre as observações, ou seja, entre instantes de tempo (EHLERS, 2005). Desse modo, uma série temporal pode ser definida como:

**Definição 1 (Série temporal)** Uma série temporal  $T = \{t_1, \dots, t_i, t_j, \dots, t_m\}$  consiste em um conjunto de  $m$  valores ordenados,  $m \geq 2$ , tal que se  $i < j$ ,  $t_i$  ocorre cronologicamente antes que  $t_j$ .

Associado a esse conceito, em muitas situações uma série temporal pode ser interpretada em função das subsequências que a compõem, sendo uma subsequência definida como:

**Definição 2 (Subsequência)** Uma subsequência  $S = \{t_p, \dots, t_{p+n-1}\}$  consiste em um subconjunto contíguo de  $n$  valores de  $T$  com início na posição  $p$ , tal que  $2 \leq n \leq m$  e  $1 \leq p \leq m - n + 1$ .

Comumente na literatura, para uma melhor compreensão e análise dos eventos representados por uma série temporal, utiliza-se o conceito de decomposição da série em um conjunto finito de componentes independentes. As principais componentes abordadas são denominadas tendência, sazonalidade e resíduo (PYLE, 1999; MORETTIN; TOLOI, 2006). Desse modo, cada uma das observações  $t_i$  que compõem uma determinada série temporal  $T$  pode estar influenciada por uma ou mais dessas componentes. No entanto, em grande parte dos problemas estudados, não é possível identificar diretamente a atuação dessas componentes na série temporal, de modo que somente podem ser extraídas e compreendidas por meio da aplicação de técnicas específicas de decomposição (BROCKWELL; DAVIS, 1996).

Geralmente as componentes de sazonalidade e tendência possuem uma forte relação, de modo que a influência da tendência sobre a componente sazonal pode afetar significativamente os métodos de análise. Nesse sentido, por meio da decomposição de uma série, pode-se isolar uma componente da outra (BROCKWELL; DAVIS, 1996; MORETTIN; TOLOI, 2006).

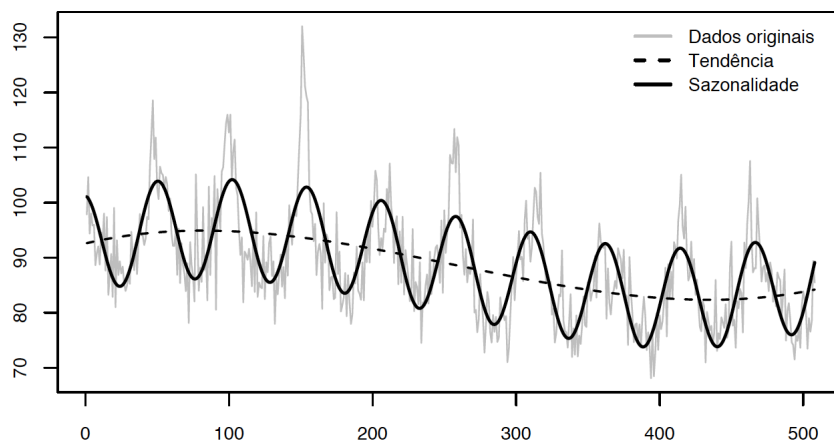


Figura 1 – Representação ilustrativa das componentes de tendência e sazonalidade (FERRERO, 2009).

**Tendência:** A componente tendência corresponde ao movimento dominante em uma série temporal, o qual exerce influência sobre as observações por longos períodos de tempo, podendo alterar o nível médio da série. Essa componente tem como característica o quase constante movimento crescente ou decrescente, que atua de modo suave ao longo da série. Na figura 1 a componente de tendência é representada por uma linha tracejada e a série temporal do fenômeno é observada em cor cinza. As séries temporais podem possuir distintos tipos de comportamentos de tendência, nos quais os métodos de identificação dessa componente estão baseados (EHLERS, 2005; SILVA, 2005). Os métodos mais utilizados nesse tipo de tarefa são: suavizar os valores da série próximos a um determinado ponto; ajustar uma função aos valores observados da série; e suavizar os valores da série por meio de sucessivos ajustes de retas de mínimos quadrados ponderados (MORETTIN; TOLOI, 2006).

**Sazonalidade:** Um determinado comportamento que tende a se repetir em uma série temporal em diferentes períodos de tempo é denominado sazonalidade. Essa componente representa as oscilações ao longo da componente de tendência de acordo com uma determinada característica (FERRERO, 2009). Dependendo do domínio de aplicação, a sazonalidade pode ser de grande interesse pois pode permitir a identificação de comportamentos relevantes do fenômeno observado ou pode dificultar a percepção de outros eventos relevantes. Desse modo, a identificação dessa componente consiste em um procedimento importante no contexto de análise de séries temporais, a qual por sua existência pode revelar informações relevantes e sua remoção pode ressaltar outras características da série temporal (MALETZKE, 2009). Na figura 1, a componente sazonalidade está representada por uma linha contínua em cor preta.

**Resíduo:** Geralmente os fenômenos caracterizados por meio de séries temporais podem apresentar eventos aleatórios característicos do próprio fenômeno ou gerados a partir dos procedimentos e equipamentos utilizados durante o processo de amostragem dos dados de interesse (MALETZKE, 2009). No contexto de análise de séries temporais, esses comportamentos são representados pela componente de resíduo. A existência dessa componente não pode ser captada pelas componentes de tendência e sazonalidade, e pode ainda dificultar a identificação das mesmas. Desse modo, a modelagem das demais componentes permite isolar e analisar o resíduo (EHLERS, 2005).

## 2.5 ANÁLISE DE SÉRIES TEMPORAIS

A análise de séries temporais consiste em uma tarefa com características específicas, as quais diferem de outros tipos de dados devido à relação de dependên-

cia existente entre as observações que compõem a série e à alta dimensionalidade frequentemente verificada nesses dados (LAXMAN; SASTRY, 2006). As abordagens tradicionais para a análise de séries temporais baseiam-se em métodos estatísticos, tais como médias móveis (MA), médias móveis exponenciais, modelos auto regressivos (AR) e a combinação de MA com AR (ARMA) (MORETTIN; TOLOI, 2006). No entanto, para que esses métodos funcionem adequadamente, é necessário assumir que os dados em análise são estacionários. Para a análise de séries não estacionárias é necessário remover alguma componente das séries temporais (CASTRO, 2012). Nessa situação, pode ser utilizado o método autoregressivo de médias móveis integrado (ARIMA), que baseia-se na remoção da componente de tendência para aproximar a série de um comportamento estacionário. Quando uma componente sazonal está envolvida, a aproximação estacionária pode ser realizada pelo modelo sazonal autoregressivo de médias móveis integrado (SARIMA) (BOX; JENKINS; REINSEL, 1994), que é uma variação do método ARIMA.

Além do requerimento de estacionariedade, os métodos estatísticos mencionados assumem que após a remoção das componentes de tendência e/ou sazonalidade, os valores do resíduo são independentes e gerados por um processo caracterizado por uma distribuição normal (EHLERS, 2005). Esses requerimentos dificilmente são satisfeitos para a maioria dos domínios de séries temporais (LAXMAN; SASTRY, 2006; CASTRO, 2012). Outros aspectos que dificultam ou impossibilitam a análise de séries temporais por meio dos métodos estatísticos tradicionais estão relacionados à necessidade de aplicação de outros tipos de tarefas de interesse. Basicamente, os métodos estatísticos foram desenvolvidos para a estimativa de valores futuros, sendo limitados ao tratamento de valores unicamente numéricos (EHLERS, 2005). Nesse contexto, com o intuito de possibilitar a análise de dados temporais que não se enquadram nas restrições das abordagens estatísticas, métodos de MD tem sido amplamente estudados na literatura nos últimos 20 anos (ANTUNES; OLIVEIRA, 2001; KEOGH; LIN; TRUPPEL, 2003; LAXMAN; SASTRY, 2006; MORCHEN, 2006; FU, 2011; ESLING; AGON, 2012).

Na próxima seção são descritas, em linhas gerais, as tarefas mais exploradas no âmbito da MD em séries temporais. Neste trabalho, essas tarefas são estruturadas em três categorias principais: tarefas relacionados ao pré-processamento de séries temporais, medidas de similaridade, e abordagens utilizando técnicas de AM em séries temporais.

## 2.6 TAREFAS DE PRÉ-PROCESSAMENTO

Considerando o processo de MD, o pré-processamento consiste em uma das tarefas mais custosas e que tem impacto direto sobre a etapa de construção de



padrões, a qual depende fortemente da qualidade dos dados utilizados (MICHALSKI; BRATKO; KUBAT, 1998; PYLE, 1999; HAN; KAMBER, 2006). No contexto de séries temporais, características como a alta dimensão dos dados e a relação existente entre as observações tornam não triviais quaisquer técnicas de transformação sobre esse tipo de dado. Nesta seção são apresentadas brevemente as tarefas comumente utilizadas na literatura para o pré-processamento de séries temporais.

### 2.6.1 Amostragem, tendência e ruído

Um problema frequentemente presente nos diversos domínios que produzem dados na forma de séries temporais, está relacionado à amostragem irregular e aos valores faltantes dos dados. Essa característica pode influenciar negativamente o desempenho das técnicas de análise que assumem a presença de todos os dados ou que os mesmos são uniformemente espaçados em relação ao tempo. Para contornar esse tipo de problema podem ser aplicados métodos de previsão, como modelos auto-regressivos (MORETTIN; TOLOI, 2006), ou métodos de interpolação (MORCHEN, 2006).

Como mencionado, a componente de tendência afeta o comportamento dominante das séries temporais e desse modo pode influenciar algumas técnicas de análise que não permitem tratar essa característica (KEOGH; PAZZANI, 1999). Para remover a componente de tendência, comumente são aplicados métodos estatísticos, tais como aqueles mencionados na seção anterior.

Outro problema geralmente encontrado para a aplicação dos métodos de análise de séries temporais refere-se à presença de ruídos. Existem diversos métodos na literatura, que foram propostos para a remoção de ruídos, sendo cada qual melhor aplicado dependendo das características do domínio. As soluções mais comuns para a remoção de ruídos consistem em aplicar técnicas de filtragem de sinais, tais como médias móveis, suavização exponencial e diferenças de primeira ordem (SHUMWAY; STOFFER, 2006; MALETZKE, 2009; CASTRO, 2012).

### 2.6.2 Normalização

A maioria das tarefas de MD por meio de métodos de AM está relacionada com a operação de comparação entre séries temporais. Como será apresentado na seção 2.7, existem distintas métricas propostas para comparar séries, no entanto, as abordagens mais utilizadas são baseadas na comparação morfológica, tais como as medidas da norma  $L_p$ . Nesse cenário, um problema comum para esse tipo de tarefa refere-se ao fato de que as séries podem estar representadas em diferentes níveis de escala. No exemplo apresentado na figura 2 (a) duas séries temporais  $U$  e  $F$  são representadas em diferentes níveis em relação ao eixo das ordenadas. Por meio de

uma análise visual é possível verificar que ambas apresentam morfologia similar, no entanto, a aplicação de medidas da norma  $L_p$  poderá indicar baixo grau de similaridade.

Uma das possíveis soluções consiste em aplicar técnicas de normalização sobre os dados das séries temporais a serem comparadas. A seguir são, brevemente, descritas três técnicas tradicionais de normalização (KEOGH; LIN, 2005): *offset*, amplitude e escala.

**Normalização de *offset*:** consiste em gerar um valor  $t'_i$  a partir da subtração de cada observação  $t_i$  de uma série temporal  $T$  pela média aritmética de todos os valores de  $t_i \in T$ . Esse processo é dado pela equação 2.1 e representado na figura 2 (b).

$$t'_i = (t_i - \mu(T)) \quad (2.1)$$

**Normalização de amplitude:** essa técnica de normalização é similar ao método de *offset*, com a diferença de que cada subtração relacionada a  $t_i$  deve ser dividida pelo desvio padrão de todos os valores de  $t_i \in T$ . Como resultado, tem-se uma série temporal  $T'$  tal que  $\mu(T') = 0$  e  $\sigma(T') = 1$ . Essa normalização pode ser obtida por meio da equação 2.2 e representado na figura 2 (c).

$$t'_i = \frac{(t_i - \mu(T))}{\sigma(T)} \quad (2.2)$$

**Normalização de escala:** nessa técnica de normalização todos os valores  $t_i \in T$  são ajustados no intervalo  $[0, 1]$ , por meio da equação 2.3. Os valores  $\min(T)$  e  $\max(T)$  correspondem ao menor e ao maior valor de  $t_i$ , respectivamente. Um exemplo de aplicação dessa técnica é apresentado na figura 2 (d)

$$t'_i = \frac{(t_i - \min(T))}{\max(T) - \min(T)} \quad (2.3)$$

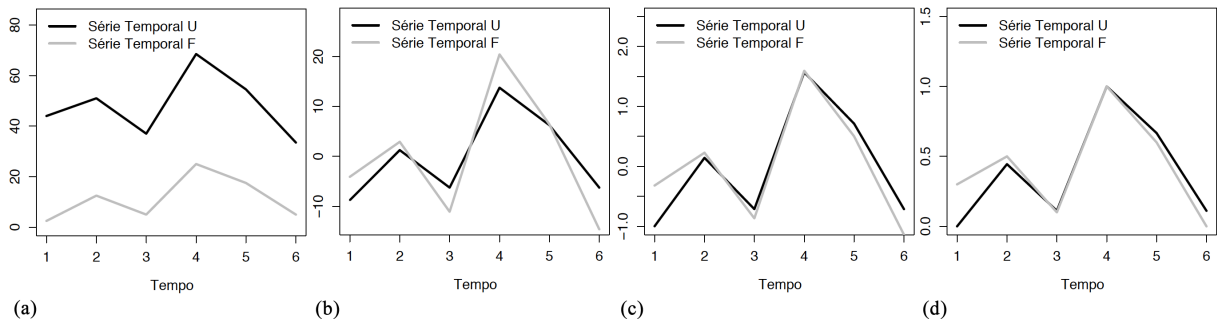


Figura 2 – Representação gráfica da aplicação das técnicas de normalização (b) *offset*, (c) amplitude e (d) escala, sobre as séries temporais (a)  $U$  e  $F$  (MALETZKE, 2009).



### 2.6.3 Representação

Devido à característica de alta dimensionalidade frequentemente encontrada nas séries temporais aliada à heterogeneidade dos dados existentes em cada domínio, a utilização de métodos de aproximação é fundamental para o correto desempenho das técnicas de análise (FALOUTSOS; RANGANATHAN; MANOLOPOULOS, 1994; LIN et al., 2003; CASTRO, 2012). Nesse contexto, a representação de séries temporais consiste em utilizar um conjunto reduzido de valores para caracterizar as séries, de modo que informações relevantes sejam preservadas ou até mesmo ressaltadas. Em geral, algumas propriedades básicas devem estar presentes, em maior ou menor grau, nos métodos de representação: capacidade de redução da dimensionalidade dos dados; preservar características locais e globais; baixo custo computacional; baixo erro de reconstrução em relação a representação original; e capacidade de tratar dados com resíduos (ESLING; AGON, 2012).

**Definição 3 (Representação)** *Seja uma série temporal  $T = \{t_1, \dots, t_m\}$  de tamanho  $m$ , uma representação  $\hat{T} = \{\hat{t}_1, \dots, \hat{t}_{m'}\}$  de tamanho  $m'$ , consiste em uma aproximação de  $T$  tal que  $m' \leq m$ .*

No âmbito da literatura de séries temporais diversas técnicas de representação têm sido propostas, cada qual com vantagens e desvantagens específicas dependendo do domínio de aplicação. De modo geral, é possível organizar essas técnicas, de acordo com o tipo de transformação utilizada (LIN et al., 2003; MORCHEN, 2006; FU, 2011; ESLING; AGON, 2012), em três categorias: adaptativas, não adaptativas e baseadas em modelos. Na figura 3 são apresentados, de modo ilustrativo, os métodos de representação mais utilizados na literatura.

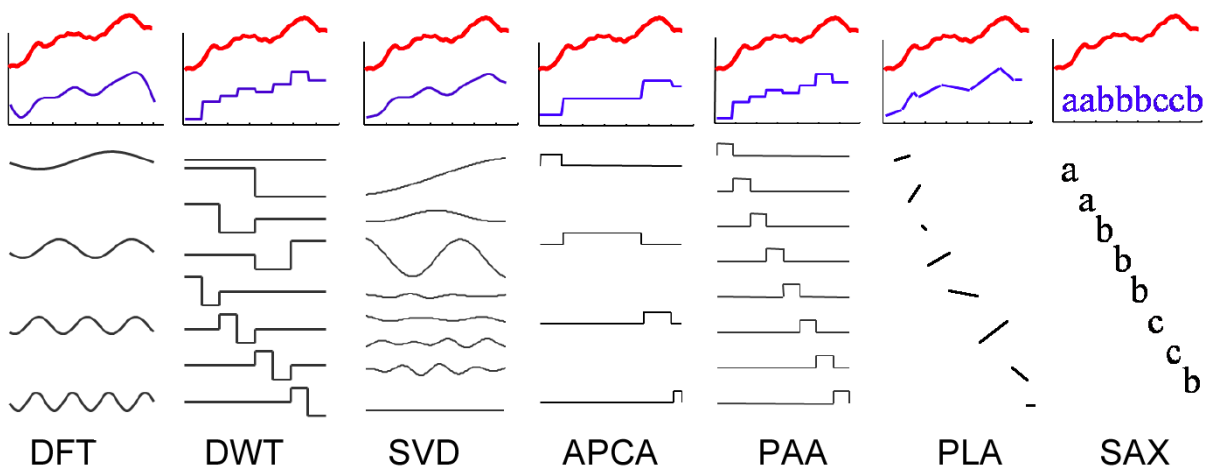


Figura 3 – Representação ilustrativa dos principais métodos da literatura de representação de séries temporais (CASTRO, 2012).

## Abordagem não-adaptativa

Os métodos baseados na abordagem não-adaptativa apresentam um processo de transformação que não leva em consideração informações dos dados das séries temporais. O método mais simples de representação dessa categoria é a amostragem (ASTROM, 1969), que consiste basicamente em selecionar  $n$  valores de uma série temporal  $T$  de tamanho  $m$ , tal que  $n \ll m$ . No entanto, esse método apresenta a desvantagem de provocar a distorção morfológica da série dependendo do valor utilizado para  $n$ . Um aprimoramento desse método é o *piecewise aggregate approximation* (PAA) (KEOGH et al., 2001a), o qual baseia-se em dividir a série em segmentos de mesmo tamanho e então utilizar a média aritmética dos valores dos segmentos consecutivos para a representação. Uma extensão desse método é o *multi-resolution piecewise aggregate approximation* (MPAA) (LIN et al., 2005), proposto para prover informações para diferentes tamanhos de segmentos.

Uma outra categoria de métodos dessa abordagem consiste em transformar as séries em outros domínios. O método *discrete fourier transform* (DFT) (AGRAWAL; FALOUTSOS; SWAMI, 1993) utiliza a decomposição do espectro de frequências para representar uma série temporal. A ideia baseia-se na representação de uma série temporal por meio da soma de uma quantidade finita de funções de seno e cosseno. Assim, nesse método são mantidos os parâmetros das funções de frequências mais baixas, enquanto que as funções de frequências mais altas são descartadas pois, em geral, são consideradas como resíduos. Uma variação desse método é o *discrete cosine transform* (DCT) (KORN; JAGADISH; FALOUTSOS, 1997), que consiste em utilizar somente funções de cosseno para a descrição da série temporal. Outro método semelhante é o *discrete wavelet transform* (DWT) (CHAN; FU, 1999), para o qual a ideia é realizar uma combinação linear de funções base *wavelets* em distintas variações de escala e *offset*. Desse modo, ao contrário dos métodos DFT e DCT, nos quais somente comportamentos periódicos globais podem ser mantidos, no método DWT é possível capturar características locais e globais da série temporal. Nesse método também são desprezados na representação final os parâmetros das funções de alta frequência. Diferentes funções *wavelets* têm sido propostas na literatura, tais como Haar (CHAN; FU; YU, 2003), Daubechies (POPIVANOV; MILLER, 2002) e Coiflets (SHASHA; ZHU, 2004).

## Abordagem adaptativa

Os métodos baseados na abordagem não-adaptativa apresentam um processo de transformação que considera alguma informação dos dados das séries temporais para produzir a representação. Em geral, a maioria dos métodos da abordagem não-adaptativa podem ser convertidos para adaptativa por meio da inclusão de uma

estratégia que considere alguma informação sobre os dados (MÖRCHEN; ULTSCH, 2005). Uma generalização do método não-adaptativo, denominado PAA, é o *adaptive piecewise constant approximation* (APCA) (GEURTS, 2001; KEOGH et al., 2001b), que consiste em flexibilizar o tamanho dos segmentos para melhor ajustar as subsequências que apresentam maior variabilidade de valores. Outro método similar é o *piecewise constant approximation* (PLA) (SHATKAY; ZDONIK, 1996), que baseia-se no ajuste de modelos lineares para cada segmento por meio de técnicas de interpolação ou regressão. O método *singular value decomposition* (SVD) (KORN; JAGADISH; FALOUTSOS, 1997) é similar aos métodos DFT e DWT no que refere-se ao uso de parâmetros de funções que caracterizam sinais, no entanto, o SVD baseia-se em um conjunto de parâmetros de funções que descrevem a variância presente nos dados da série temporal. A estratégia do método *perceptually important points* (PIPs) é preservar na representação somente as observações da série temporal que forem consideradas importantes em relação aos valores vizinhos (BAO, 2008). O método *derivative segment approximation* (DSA) (GULLO et al., 2009) consiste em transformar a série por meio das diferenças de primeira ordem e após dividir em segmentos que permitam representar a informação de inclinação aproximada de cada um. Recentemente, em Lines et al. (2012) os autores propuseram representar uma série temporal por meio de um conjunto de subsequências discriminantes, denominadas *shapelets* (YE; KEOGH, 2009). Essa abordagem é detalhada no próximo capítulo.

Considerando que algumas técnicas tradicionais de análise de dados temporais foram desenvolvidas apenas para processar dados discretos (AGRAWAL; SRIKANT, 1995; MANNILA; TOIVONEN; VERKAMO, 1997), uma outra categoria de métodos da abordagem adaptativa consiste em produzir uma representação simbólica das séries temporais. Além desse aspecto, outra motivação para o desenvolvimento desses métodos é a possibilidade de prover um maior nível de inteligibilidade do conhecimento representado nas séries temporais (ALONSO et al., 2008; MÖRCHEN; ULTSCH, 2005). Nesse contexto, diversas estratégias tem sido propostas na literatura. O método de distribuição uniforme consiste em representar os valores de uma série temporal por meio de histogramas igualmente espaçados. De modo semelhante, o método de distribuição por frequências baseia-se na construção de histogramas que contenham a mesma quantidade de valores (DAW; FINNEY; TRACY, 2003). Em ambos os métodos, um símbolo é atribuído para cada valor da série de acordo com o histograma ao qual estão contidos. O algoritmo *PERSIST* (MÖRCHEN; ULTSCH, 2005) foi proposto para identificar automaticamente os limites de cada histograma levando em consideração a ordem temporal dos símbolos. O método de discretização mais utilizado na literatura atual é denominado *symbolic aggregate approximation* (SAX) (LIN et al., 2007). Esse método utiliza uma etapa de pré-processamento por meio da aplicação do método PAA e assume que valores gerados por essa representação seguem uma distribuição normal.

Assim, os valores da série resultante são estruturados em histogramas, que têm seus limites definidos segundo a função densidade da distribuição normal. Ao final, cada valor da série é associado ao símbolo do histograma correspondente. Uma proposta baseada na utilização do método PAA, denominada *piecewise vector quantized approximation* (PVQA), foi apresentada em [Megalooikonomou, Li e Wang \(2004\)](#) para criar um dicionário de palavras para representar cada segmento da série.

O método *shape description alphabet* (SDA) ([AGRAWAL et al., 1995](#)) utiliza a informação de inclinação entre as observações adjacentes para produzir símbolos que representam transições pré-definidas, tais como crescente, estável e decrescente. Posteriormente em [Qu, Wang e Wang \(1998\)](#), essa estratégia foi generalizada pela divisão da série em segmentos. De modo semelhante, em [Huang e Yu \(1999\)](#) é proposto o método IMPACTS, no qual valores iguais e consecutivos de inclinação definem os segmentos que serão discretizados. Em [Giles, Lawrence e Tsoi \(2001\)](#) os autores utilizam *self-organizing maps* para criar símbolos com base na divisão da série em segmentos de diferenças de primeira ordem. Uma outra técnica, denominada *clipping* ([RATANAMAHATANA et al., 2005](#); [BAGNALL et al., 2006](#)) consiste em representar séries temporais por meio sequências de valores binários, denominados *bits*. Cada *bit* indica que um dado valor está acima ou abaixo da média (ou mediana) dos valores da série. Em ([DAS et al., 1998](#); [HUGUENEY](#); [BOUCHON-MEUNIER, 2001](#)) os autores propuseram a aplicação de algoritmos de agrupamento sobre subsequências para a definição de símbolos.

#### *Contribuições deste trabalho relacionadas à representação de séries temporais*

Algumas das contribuições deste trabalho de doutorado tiveram como foco o desenvolvimento de métodos para a representação simbólica de séries temporais. A seguir são, brevemente, descritas as publicações realizadas nesse contexto:

1. *A Symbolic Representation Method to Preserve the Characteristic Slope of Time Series* ([ZALEWSKI et al., 2012a](#)): nesse trabalho foi proposto um método de discretização para preservar a informação de inclinação local entre os valores que compõem as séries temporais;
2. *Time Series Discretization Based on the Approximation of the Local Slope Information* ([ZALEWSKI et al., 2012b](#)): nesse trabalho consiste em uma extensão do trabalho anterior ([ZALEWSKI et al., 2012a](#)), no qual foi introduzida uma abordagem baseada em janela deslizando combinada com o cálculo das diferenças de primeira ordem;
3. *Symbolic Representation Based on Temporal Order Information for Time Series Classification* ([ZALEWSKI et al., 2013](#)): nesse trabalho foi proposto um método

de representação por meio de descritores de características simbólico-temporais.

### Abordagem baseada em modelos

Os métodos baseados nessa abordagem assumem que os valores que compõem uma série temporal são gerados por um modelo base. Com isso, a representação é realizada por meio dos parâmetros de ajuste para um determinado modelo de aproximação. Os métodos mais comuns dessa abordagem incluem a utilização de modelos estatísticos, como o ARMA (KALPAKIS; GADA; PUTTAGUNTA, 2001); cadeias de Markov para séries temporais simbólicas (SEBASTIANI; RAMONI, 2001); e *hidden Markov models* (HMM) (PANUCCIO; BICEGO; MURINO, 2002). Os métodos que utilizam a extração de características estatísticas também podem ser interpretados como um modelo base do processo gerador das séries (NANOPOULOS; ALCOCK; MANOLOPOULOS, 2001; WANG; HAN, 2004). Recentemente, em Silva, Souza e Batista (2013) os autores propuseram representar as séries temporais por meio de modelos, que podem ser interpretados como imagens, denominados *recurrence plots*.

## 2.7 MEDIDAS DE SIMILARIDADE

A noção de similaridade consiste em um aspecto fundamental para a análise de séries temporais, em especial para a maioria das tarefas de MD. Formalmente uma medida de similaridade pode ser definida como:

**Definição 4 (Medida de similaridade)** *Uma medida de similaridade consiste em uma função  $Dist$  que dadas duas séries temporais  $T_x = \{x_1, x_2, \dots, x_m\}$  e  $T_y = \{y_1, y_2, \dots, y_m\}$  como entrada, retorna um valor numérico:*

$$Dist : T_x \times T_y \rightarrow \mathbb{R} \quad (2.4)$$

Uma grande quantidade de métricas de similaridade tem sido proposta na literatura, cada qual com características distintas. Desse modo, a escolha de uma medida de similaridade adequada depende da natureza dos dados a serem analisados, dos requisitos específicos de cada domínio de aplicação, e frequentemente, do tipo de representação adotada (MORCHEN, 2006). De modo geral, as medidas de similaridade devem prover resultados consistentes com a intuição humana; ressaltar comportamentos locais e globais; abstração de distorções e invariância a transformações. Adicionalmente, as medidas de similaridade de séries temporais podem ser divididas em quatro principais categorias (MORCHEN, 2006; ESLING; AGON, 2012): baseadas na forma, baseadas em características, baseadas na edição e baseadas na estrutura.

### 2.7.1 Medidas baseadas na forma

As medidas baseadas na forma apresentam como característica a capacidade de prover uma informação mais próxima da análise visual, pois tem como objetivo a descrição de aspectos morfológicas das séries temporais (CASTRO, 2012). Nessa categoria, as medidas frequentemente utilizadas na literatura são as medidas da norma  $L_p$  (YI; FALOUTSOS, 2000), em especial a distância Euclidiana (ED),  $p = 2$  (KEOGH; KASETTY, 2003). No entanto, essas medidas apresentam algumas restrições relacionadas à comparação de séries em diferentes níveis de amplitude em termos de escala e translação, para as quais é necessária a aplicação de técnicas de normalização. Outras deficiências referem-se à necessidade de que as séries temporais a serem comparadas tenham o mesmo tamanho e não apresentem comportamentos de interesse defasados no tempo. Para contornar esses problemas, medidas como a *dynamic time warping* (DTW) (SAKOE; CHIBA, 1978) têm sido amplamente exploradas na literatura de séries temporais (BERNDT; CLIFFORD, 1994; KEOGH; PAZZANI, 1998; KEOGH; PAZZANI, 2001; KEOGH; RATANAMAHATANA, 2005; SALVADOR; CHAN, 2007; FU et al., 2008; JEONG; JEONG; OMITAOMU, 2011). Recentemente, em Batista, Wang e Keogh (2011) foi proposta a medida *complexity invariant distance* (CID) para tratar a invariância à complexidade, a qual pode ser utilizada em combinação com ED e DTW.

Considerando que neste trabalho as medidas de distância Euclidiana e DTW são utilizadas pelos algoritmos propostos e pelos algoritmos avaliados, a seguir é apresentada uma breve descrição de cada uma dessas medidas.

#### Distância Euclidiana

Essa medida determina qual a distância em linha reta entre dois pontos, os quais pertencem a um espaço de  $m$  dimensões, onde  $m$  corresponde ao tamanho de uma determinada série temporal. A distância Euclidiana  $Dist_{ED}$  entre duas séries temporais  $T_x$  e  $T_y$ , é definida conforme a Equação 2.5:

$$Dist_{ED}(T_x, T_y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2} \quad (2.5)$$

onde  $i$  indica cada dimensão em cada iteração para o cálculo da distância.

#### Dynamic Time Warping

Diferentemente da medida ED, a DTW permite ajustar o melhor alinhamento entre duas séries temporais, com o intuito de tratar possíveis distorções.

Para determinar a similaridade por meio da DTW, entre duas séries temporais  $T_x$  e  $T_y$  de tamanho  $m_x$  e  $m_y$ , é construída uma matriz  $m_x \times m_y$ , na qual cada elemento



$(i, j)$  dessa matriz contém a distância  $Dist_{ED}(x_i, y_j)$  entre dois pontos  $x_i$  e  $y_j$ . Cada elemento  $(i, j)$  da matriz corresponde ao alinhamento entre os pontos  $x_i$  e  $y_j$ . Essa situação é ilustrada na Figura 4, na qual uma rota  $W = \{w_1, w_2, \dots, w_k, \dots, w_K\}$  consiste em um conjunto contíguo de elementos da matriz, tal que o  $k$ -ésimo elemento de  $W$  é definido como  $w_k = Dist_{ED}(x_i, y_j)$ ,  $\max(m_x, m_y) \leq K < m_x + m_y - 1$ . A determinação da rota  $W$ , que define o alinhamento entre  $T_x$  e  $T_y$  está, geralmente, sujeita a várias restrições:

- Condições de fronteira:  $w_1 = (1, 1)$  e  $w_K = (m_x, m_y)$ , indica que a rota começa e termina na posição diagonalmente oposta da matriz;
- Continuidade: dado  $w_k = (a, b)$  então  $w_{k-1} = (a', b')$ , onde  $a - a' \leq 1$  e  $b - b' \leq 1$ . Isso restringe os passos na rota para posições adjacentes (incluindo a diagonal adjacente);
- Monotonicidade: dado  $w_k = (a, b)$  então  $w_{k-1} = (a', b')$ , onde  $a - a' \geq 0$  e  $b - b' \geq 0$ , restringindo os pontos em  $W$  a serem uniformemente espaçados no tempo.

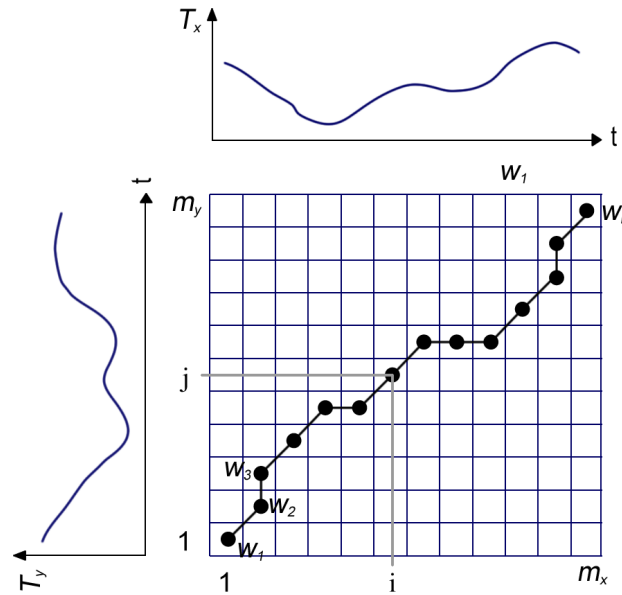


Figura 4 – Exemplo de uma matriz e da rota  $W$  obtidas a partir da aplicação da DTW (CHU et al., 2002).

Podem existir diversas rotas que satisfazem as condições mencionadas, no entanto, busca-se unicamente a rota que minimiza o custo de alinhamento:

$$Dist_{DTW}(T_x, T_y) = \min \left\{ \frac{1}{K} \sqrt{\sum_{k=1}^K w_k} \right\} \quad (2.6)$$

A variável  $K$  no denominador é usada para compensar o fato de que a rota  $W$  pode apresentar diferentes tamanhos. Essa rota pode ser encontrada utilizando

programação dinâmica para avaliar as próximas repetições que definem a distância acumulativa  $\gamma(i, j)$  como a distância  $Dist_{ED}(i, j)$  encontrada na posição atual e a mínima distância acumulativa dos elementos adjacentes, dada pela Equação 2.7:

$$\gamma(i, j) = Dist_{ED}(x_i, y_j) + \min \{ \gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \} \quad (2.7)$$

Na figura 5 é representada a diferença de alinhamento entre as medidas de similaridade ED e DTW.

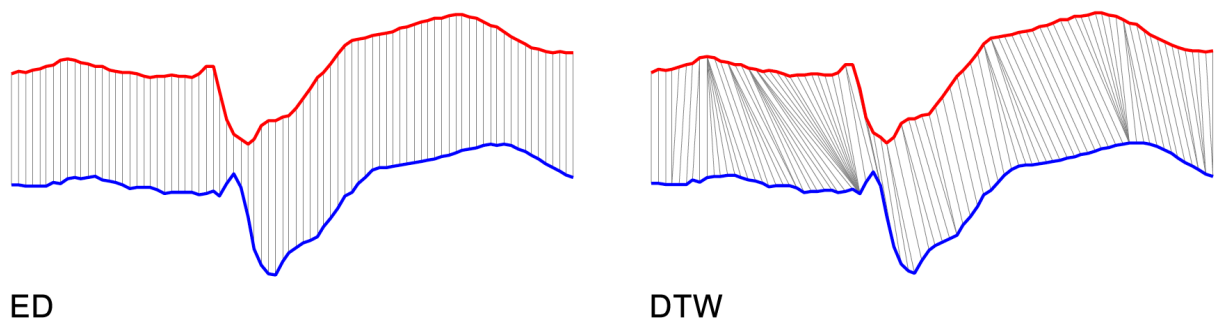


Figura 5 – Representação ilustrativa da diferença de alinhamento entre as distâncias Euclidiana e DTW (SILVA; SOUZA; BATISTA, 2013).

De modo geral, as medidas baseadas na forma são mais adequadamente aplicadas quando a análise envolver séries temporais de tamanho relativamente pequenos e se comportamentos morfológicos são descritores importantes (MORCHEN, 2006; ESLING; AGON, 2012).

### 2.7.2 Medidas baseadas em características

As medidas de similaridade baseadas nessa abordagem são aplicadas sobre representações que permitam descrever a série temporal utilizando um conjunto de características. Desse modo, a similaridade entre duas séries é determinada por meio dos valores das características selecionadas. Como apresentado na seção anterior, DFT e DWT são exemplos de métodos que produzem representações baseadas em características, nesse caso por meio de parâmetros de funções. Uma das abordagens existentes na literatura consiste em utilizar a distância Euclidiana sobre os parâmetros dessas representações (AGRAWAL; FALOUTSOS; SWAMI, 1993; CHAN; FU, 1999). Outras abordagens incluem a utilização da taxa de probabilidade (*likelihood ratio*) sobre a representação DFT (JANACEK; BAGNALL; POWELL, 2005); e a aplicação de periodogramas e funções de auto-correlação (VLACHOS; YU; CASTELLI, 2005).

Em geral, as medidas baseadas em características são mais apropriadas em domínios de séries temporais de tamanhos relativamente grandes, nas quais determinados comportamentos devem ser ressaltados enquanto outros podem ser descartados (MORCHEN, 2006; ESLING; AGON, 2012).



### 2.7.3 Medidas baseadas na edição

As medidas dessa abordagem foram originalmente propostas para a comparação entre sequências de símbolos. O método básico, chamado de distância de edição, consiste em determinar a distância entre duas séries temporais  $T_x$  e  $T_y$  por meio da quantidade de operações (inserção, substituição, remoção) mínimas necessárias para transformar uma delas na outra. A medida *longest common subsequence* (LCSS) (DAS; GUNOPULOS; MANNILA, 1997; VLACHOS; GUNOPOULOS; KOLLIOS, 2002) é um dos métodos mais conhecidos dessa abordagem. A similaridade por meio da LCSS é determinada com base no tamanho da maior subsequência em comum entre  $T_x$  e  $T_y$ . Dentre diversas abordagens existentes, citam-se a *edit distance on real sequence* (EDR) (CHEN, 2005) que é uma adaptação da distância de edição para tratar séries temporais numéricas; e as medidas *edit distance with real penalty* (EPR) (CHEN; NG, 2004) e *time warp edit distance* (TWED) (MARTEAU, 2009) que foram propostas para combinar as vantagens da DTW e da distância de edição.

### 2.7.4 Medidas baseadas na estrutura

Nessa abordagem as séries temporais são comparadas por meio de métricas probabilísticas relacionadas à estrutura global das séries. A ideia básica para a determinação do grau de similaridade entre duas séries temporais  $T_x$  e  $T_y$  consiste em construir um modelo base para caracterizar  $T_x$  como referência paramétrica. Assim, a similaridade das duas séries é expressa pela probabilidade de  $T_y$  ser gerada pelo mesmo modelo base de  $T_x$ . Como proposto em Esling e Agon (2012), as medidas dessa abordagem podem ser organizadas em duas categorias. A primeira caracteriza os métodos baseados no ajuste de modelos paramétricos temporais, tais como HMM e ARMA (PANUCCIO; BICEGO; MURINO, 2002; XIONG; YEUNG, 2004), e modelos de regressão (GAFFNEY; SMYTH, 1999). A segunda categoria é caracterizada pela medida *compression-based dissimilarity measure* (CDM) (KEOGH; LONARDI; RATANAMAHATANA, 2004; KEOGH et al., 2007), a qual é baseada no conceito da complexidade de Kolmogorov. Assim, a motivação da medida CDM é que concatenar e compactar séries temporais similares deve produzir taxas de compressão maiores do que para as séries não similares. As medidas baseadas na estrutura são frequentemente indicadas quando a análise envolve a identificação de similaridades de modo global, e existe algum conhecimento *a priori* sobre o processo gerador das séries temporais (MORCHEN, 2006; ESLING; AGON, 2012).

## 2.8 APRENDIZADO DE MÁQUINA EM SÉRIES TEMPORAIS

Como mencionado, devido às limitações dos métodos estatísticos para a análise de séries temporais, diversas abordagens baseadas em técnicas de AM foram

propostas nos últimos anos. Nesta seção são brevemente descritos os principais tipos de análise desenvolvidos por meio de métodos de AM, com especial destaque para a tarefa de classificação, a qual é o foco deste trabalho.

### 2.8.1 Recuperação por conteúdo

Seja um conjunto de séries temporais, a tarefa de recuperação por conteúdo apresenta como principal objetivo a consulta pela existência de uma determinada série temporal ou de suas similares (ESLING; AGON, 2012). Os principais aspectos envolvidos nos métodos propostos para essa tarefa estão relacionados à eficiência do processo de busca, em termos do custo de tempo e da qualidade dos resultados das consultas (grau de similaridade) (MORCHEN, 2006). Nesse contexto, os fatores determinantes referem-se aos métodos adotados para a representação das séries temporais e/ou das medidas de similaridade. Em Hetland (2004) são descritas diferentes abordagens de recuperação por conteúdo, bem como os critérios importantes que devem ser considerados ao se aplicar essa tarefa.

### 2.8.2 Agrupamento

Dado um conjunto de séries temporais, a tarefa de agrupamento consiste em definir um conjunto de grupos, de modo que a similaridade entre as séries de um mesmo grupo sejam maximizadas, enquanto que a similaridade entre as séries de grupos diferentes possam ser minimizadas. No contexto de agrupamento de séries temporais, muitas abordagens tem sido propostas (VLACHOS et al., 2003; JANACEK; BAGNALL; POWELL, 2005; CORDUAS; PICCOLO, 2008). Em geral, definindo-se alguma abordagem para a representação das séries e uma medida de similaridade, os métodos de agrupamento tradicionais podem ser aplicados (HAN; KAMBER, 2006). Uma revisão dos métodos de agrupamento de séries temporais é apresentada em Liao (2005) e Rani e Sikka (2012).

### 2.8.3 Descoberta de regras

A descoberta de regras é uma tarefa amplamente conhecida de AM. Em especial, os métodos baseados na construção de regras possibilitam a representação do conhecimento por meio de abstrações que apresentam mais proximidade com o raciocínio lógico adotado pelos seres humanos (FACELI et al., 2011). Um dos algoritmos mais difundidos é o de identificação de regras de associação (AGRAWAL; IMIELIŃSKI; SWAMI, 1993; AGRAWAL; SRIKANT, 1994), que baseia-se no aprendizado não-supervisionado. No contexto de séries temporais, é necessária a aplicação de algum tipo de transformação, pois os algoritmos foram desenvolvidos para tratar sequências simbólicas (SRIKANT; AGRAWAL, 1996). Dentre as abordagens propostas

para a busca por regras de associação em série temporais (LAST; KLEIN; KANDEL, 2001; HöPPNER, 2001; MORCHEN, 2006), um trabalho inicial foi apresentado em Das et al. (1998), no qual os autores utilizaram o agrupamento de subsequências para a discretização das séries temporais.

#### 2.8.4 Previsão de valores

A previsão de valores é uma das tarefas mais estudadas no contexto de séries temporais (MORETTIN; TOLOI, 2006; MORCHEN, 2006; ESLING; AGON, 2012). Seja uma série temporal  $T = \{t_1, \dots, t_m\}$ , essa tarefa consiste em prever uma quantidade de  $m'$  valores futuros  $\{t_{m+1}, \dots, t_{m+m'}\}$  a partir de  $T$ . Os métodos tradicionais utilizados para a previsão de valores são baseados na construção de modelos estatísticos, tais como ARIMA e SARIMA (BOX; JENKINS; REINSEL, 1994; MORETTIN; TOLOI, 2006). No entanto, como mencionado na seção 2.5, as restrições relacionadas a esses métodos motivaram o desenvolvimento de abordagens baseadas em outras técnicas, tais como as de AM. Nesse contexto, algumas abordagens propuseram adaptar os algoritmos de aprendizado para a tarefa de previsão, tais como redes neurais artificiais (KOSKELA, 2003; YADAV; KALRA; JOHN, 2007), SVM (HERRERA et al., 2007), agrupamento (SFETSOS; SIRIOPOULOS, 2004), e kNN (FERRERO, 2009). Em Ahmed et al. (2010) é apresentada uma revisão e uma avaliação experimental de diversos métodos, na qual os autores apontam modelos de regressão Gaussianos e redes neurais como os melhores métodos para essa tarefa.

#### 2.8.5 Identificação de eventos

Essa tarefa tem apresentado um grande interesse dos pesquisadores de séries temporais nos últimos anos. Duas principais subtarefas podem ser relacionadas, a detecção de anomalias e a identificação de *motifs*. A detecção de anomalias (*discords*) (KEOGH et al., 2006) em uma série temporal consiste em identificar eventos que raramente ocorrem. Em geral, esses eventos são caracterizados por subsequências que destoam do comportamento geral da série temporal. O método tradicional para a identificação desses eventos baseia-se na definição de um modelo que caracterize os eventos considerados comuns na série temporal. Desse modo, as subsequências que não forem adequadamente ajustadas ao modelo, segundo algum critério, são consideradas anomalias. Uma revisão sobre os principais métodos propostos para essa sub tarefa é apresentada em Weiss (2004) e em Chandola, Banerjee e Kumar (2009).

A identificação de *motifs* é derivada da análise do sequenciamento genético da área de bioinformática, e consiste na identificação de eventos recorrentes ao longo da série temporal. Em geral, o processo de identificação de *motifs* em séries temporais refere-se à busca por subsequências não sobrepostas que ocorrem sob um

determinado limiar de frequência (PATEL et al., 2002). Nos trabalhos de Chiu, Keogh e Lonardi (2003), Liu et al. (2005), Yankov et al. (2007), Mueen et al. (2009) são descritas diferentes abordagens para a realização dessa tarefa. Em Castro (2012) os autores apresentam uma revisão sobre os principais métodos existentes na literatura. Uma variação do conceito de *motifs* foi proposta em Ye e Keogh (2009) para a identificação de subsequências discriminantes, denominada *shapelet*. Essa abordagem é descrita em maiores detalhes no próximo capítulo.

#### 2.8.6 Classificação de séries temporais

A tarefa de classificação é uma das mais estudadas na literatura de séries temporais, com início há aproximadamente duas décadas (BAKSHI et al., 1994), essa tarefa consiste em determinar uma função que permita associar uma classe a uma série temporal não rotulada. Formalmente, seja  $\mathbb{T}$  o conjunto de todas as séries possíveis de um determinado domínio e seja  $C = \{c_1, \dots, c_w\}$  um conjunto de  $w$  classes, tal que:

$$\forall T_i \in \mathbb{T} : ((T_i \in c_1) \vee \dots \vee (T_i \in c_w)) \wedge (T_i \in c_j \rightarrow T_i \notin c_k, j \neq k) \quad (2.8)$$

um classificador de séries temporais consiste em uma função  $f$  que permite mapear uma série  $T_i \in \mathbb{T}$  para uma classe  $c \in C$ :

$$f : \mathbb{T} \rightarrow \{c_1, \dots, c_w\} \quad (2.9)$$

Os métodos propostos na literatura para a classificação de séries temporais são fortemente relacionados ao tipo de representação utilizada. Nesse sentido, duas estratégias gerais tem sido adotadas. A primeira baseia-se na utilização dos dados originais das séries, para a qual a adaptação dos algoritmos tradicionais de aprendizado ou a definição de uma medida de similaridade tem fundamental importância. Na segunda estratégia, as séries temporais são transformadas para uma representação de características, as quais são utilizadas para a indução de classificadores por meio de algoritmos tradicionais de AM (MORCHEN, 2006). No estudo elaborado por Bagnall et al. (2012), os autores sugerem que em domínios nos quais os comportamentos importantes são melhores caracterizados por similaridades baseadas na forma e em modelos, a transformação dos dados originais das séries para uma representação alternativa possibilita melhor desempenho em termos de acurácia.

Dentre os métodos baseados na extração de características, um trabalho inicial foi apresentado em Bakshi et al. (1994), no qual os autores propuseram a representação de séries temporais por meio de formas pré-definidas relacionadas ao comportamento de processos químicos. Utilizando essa representação foram induzidos classificadores

baseados em árvores de decisão. Em [Manganaris \(1995\)](#), as séries foram divididas em segmentos, para os quais modelos de ajuste polinomiais foram aplicados e utilizando os parâmetros desses ajustes como representação, modelos de classificação foram induzidos por meio do algoritmo de aprendizado *Naive Bayes*. No trabalho de [Keogh e Pazzani \(1998\)](#) os autores representaram as séries temporais por meio do método PLA e apresentaram um classificador chamado *cluster then classify*.

Em [Deng, Moore e Nechyba \(1997\)](#), as séries foram representadas por meio dos parâmetros de ajuste de modelos ARMA. No trabalho de [Zhong e Ghosh \(2002\)](#) foi utilizada a representação por meio dos parâmetros de modelos HMM. Em [Zhang, Ho e Lin \(2004\)](#) os coeficientes da aplicação da DWT foram utilizados para estimar a escala a ser adotada para a representação das séries. Nesses estudos, a classificação foi realizada por meio do algoritmo 1NN em combinação com a distância Euclidiana.

Redes neurais artificiais foram aplicadas sobre a representação dos parâmetros da DWT ([ROVERSO, 2000](#)) e sobre a representação de descritores de características estatísticas ([NANOPOULOS; ALCOCK; MANOLOPOULOS, 2001](#); [WANG; HAN, 2004](#)). No trabalho de [Rodríguez, Alonso e Maestro \(2005\)](#) as séries foram divididas em intervalos, para os quais foram determinados descritores binários baseados em valores de média e desvio padrão. Essa representação é utilizada para a construção de modelos com SVM. Em [Deng et al. \(2013\)](#) os autores também utilizam a divisão da série em intervalos, para os quais descritores estatísticos como média, desvio padrão e inclinação são utilizados como entrada para o algoritmo de aprendizado de florestas aleatórias.

Em [Maletzke et al. \(2009\)](#) os autores propuseram representar as séries temporais por meio de descritores estatísticos como média, variância, obliquidade; em combinação com abordagens de identificação de *motifs*. Posteriormente utilizaram os algoritmos 1NN e J48 para a construção de modelos de classificação. No trabalho de [Buza e Schmidt-Thieme \(2010\)](#) os autores utilizaram a quantidade de ocorrências de *motifs* como característica fundamental para a indução de classificadores com redes bayesianas e SVM. Com ideia similar, outros métodos adaptaram abordagens de recuperação de conteúdo, como *bag-of-words*, para descrever a frequência de determinadas subsequências nas séries temporais. Em [Lin e Li \(2009\)](#), [Lin, Khade e Li \(2012\)](#) foi apresentado o método *bag-of-patterns* (BoP), que utiliza o algoritmo SAX para a discretização das séries temporais e o algoritmo 1NN para a classificação. Na mesma linha, em [Baydogan, Runger e Tuv \(2013\)](#) os autores apresentam o método de representação *time series bag-of-features* (TSBF), que combina a extração de descritores estatísticos com o método BoP. Nessa abordagem os autores utilizaram os algoritmos de aprendizado de árvores aleatórias e SVM. Outra variação do método BoP consiste em criar um único conjunto de símbolos para cada classe ([SENIN; MALINCHIK, 2013](#)).

Nessa abordagem os autores propuseram a aplicação da técnica *vector space model* (VSM) baseada na pontuação ponderada  $tf*idf$ . A classificação é realizada por meio do algoritmo 1NN utilizando a medida de similaridade cosseno.

Em [Silva, Souza e Batista \(2013\)](#) os autores propuseram uma representação das séries temporais por meio de imagens (*recurrence plots*). Para a classificação de novos exemplos foi proposta a aplicação do algoritmo 1NN em combinação com a medida CK-1 ([CAMPANA; KEOGH, 2010](#)). Utilizando essa mesma representação, em [Souza, Silva e Batista \(2014\)](#) os autores propuseram a construção de modelos por meio do algoritmo SVM.

Em relação aos métodos que baseiam-se na representação original dos dados das séries temporais, a tarefa fundamental consiste no tratamento da variabilidade intra-classe, a qual é caracterizada pela existência de comportamentos defasados no tempo, distorcidos e em diferentes níveis de escala. Nesse contexto, algumas abordagens propuseram adaptar os algoritmos de aprendizado tradicionais. Como exemplo, na construção de classificadores utilizando SVM, as funções núcleo clássicas não são adequadas para tratar as invariâncias necessárias ([GUDMUNDSSON; RUNARSSON; SIGURDSSON, 2008](#)). Desse modo, em [Zhang et al. \(2010\)](#) foi proposta uma função núcleo baseada na medida DTW, denominada *Gaussian DTW* (GDTW). Alternativamente, no estudo de [Grabocka, Nanopoulos e Schmidt-Thieme \(2012\)](#) os autores propuseram adicionar novos exemplos ao conjunto de treinamento, que permitissem caracterizar diferentes tipos de variações.

A abordagem amplamente explorada na literatura consiste na utilização do algoritmo 1NN com alguma medida de similaridade que permita o tratamento das variações intra-classe. Diversas outras medidas de similaridade tem sido desenvolvidas com esse propósito, tais como LCSS, EDR, ERP, CID, *move-split-merge* (MSM) ([STEFAN; ATHITSOS; DAS, 2013](#)), *weighted DTW* (WDTW) ([JEONG; JEONG; OMITAOMU, 2011](#)), *derivative DTW* (DDTW) ([KEOGH; PAZZANI, 2001](#)). Em [Giusti e Batista \(2013\)](#) e em [Serra e Arcos \(2014\)](#) são avaliadas experimentalmente diversas medidas de similaridade. Contudo, o algoritmo 1NN em combinação com a medida DTW (1NN-DTW) ainda é considerado o método estado da arte para a classificação de séries temporais, em termos do desempenho de acurácia. ([DING et al., 2008](#); [WANG et al., 2013](#); [BATISTA et al., 2013](#)).

Como discutido em [Yamada et al. \(2003\)](#), [Yamada et al. \(2005\)](#), [Ye e Keogh \(2009\)](#), [Ye e Keogh \(2011\)](#), em domínios que envolvem processos de auxílio à tomada de decisão, é de fundamental importância prover um entendimento sobre raciocínio utilizado pelo classificador ([XING et al., 2011](#); [XING; PEI; YU, 2012](#); [GHALWASH; OBRADOVIC, 2012](#); [GHALWASH; RADOSAVLJEVIC; OBRADOVIC, 2013](#); [GHALWASH; RADOSAVLJEVIC; OBRADOVIC, 2014](#)). No entanto, os classificadores baseados em



abordagens *lazy*, tais como o 1NN, não apresentam um modelo explícito do conhecimento *a priori*. Além desse aspecto, no algoritmo 1NN o resultado do processo de classificação corresponde ao grau de similaridade com o exemplo mais próximo na base de séries temporais. Essa característica pode prejudicar a interpretação dos resultados, pois possíveis comportamentos de interesse locais são generalizados em um único valor global. Em relação aos demais métodos de classificação mencionados nesta seção, a maioria possibilita a construção de modelos, no entanto não inteligíveis. Uma alternativa comumente adotada para atingir essa característica tem sido a construção de classificadores baseados no aprendizado simbólico (BAKSHI et al., 1994; KADOUS, 1999; KADOUS; SAMMUT, 2004; KADOUS; SAMMUT, 2005; KUDO; TOYAMA; SHIMBO, 1999; COTOFREI; STOFFEL, 2002; RODRIGUEZ; ALONSO; BOSTROM, 2001; COTOFREI; STOFFEL, 2005). Entretanto, a maioria dos métodos da literatura de séries temporais basearam-se principalmente na utilização de descritores de características estatísticas e/ou parâmetros de ajuste de funções, os quais são mais distantes da percepção humana. Nesse contexto, uma abordagem emergente, proposta para a descrição de atributos morfológicos e de modelos inteligíveis foi apresentada em Ye e Keogh (2009), denominada primitiva *shapelet*, que será detalhada no próximo capítulo, devido ao fato de ser utilizada como técnica base neste trabalho.

## 2.9 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os principais aspectos referentes à mineração de dados em séries temporais por meio da caracterização de tarefas como pré-processamento, medidas de similaridade, e abordagens baseadas em técnicas de aprendizado de máquina. Em especial, foram relacionados os principais métodos de classificação de séries temporais propostos na literatura, por ser a tarefa de interesse deste trabalho. Nesse contexto, apesar dos diversos métodos existentes na literatura, ainda poucas abordagens foram propostas com o intuito de prover modelos simbólicos de classificação que permitissem resultados comparáveis ao algoritmo 1NN-DTW. Desse modo, no próximo capítulo são descritas as estratégias baseadas no conceito da primitiva *shapelet* (YE; KEOGH, 2009), a qual tem demonstrado promissor desempenho em termos da acurácia e da inteligibilidade dos modelos de classificação.

### 3 INTELIGIBILIDADE NA CLASSIFICAÇÃO DE SÉRIES TEMPORAIS

#### 3.1 CONSIDERAÇÕES INICIAIS

Como descrito no capítulo anterior, a tarefa de classificação de séries temporais tem sido amplamente explorada na literatura. Nesse contexto, o algoritmo de aprendizado baseado em exemplos, 1-vizinho mais próximo, tem sido apontado como a melhor abordagem para uma grande variedade de domínios de séries temporais, em termos da acurácia de classificação (XI et al., 2006; DING et al., 2008; BATISTA et al., 2013; WANG et al., 2013). Entretanto, em algumas aplicações reais, especialmente as que estejam relacionadas a processos de auxílio à tomada de decisões (exemplos: medicina, saúde e sistemas de monitoração de segurança), é necessário expor o raciocínio realizado pelo classificador de modo inteligível. Nesse cenário, o foco não deve estar somente na acurácia do modelo, mas também na sua inteligibilidade (BATYRSHIN; SHEREMETOV, 2008; XING et al., 2011; XING; PEI; YU, 2012; GHALWASH; OBRADOVIC, 2012; GHALWASH; RADOSAVLJEVIC; OBRADOVIC, 2013; GHALWASH; RADOSAVLJEVIC; OBRADOVIC, 2014).

Como mencionado no capítulo anterior, os algoritmos baseados no paradigma de aprendizado de máquina simbólico têm a capacidade de prover uma representação do conhecimento por meio de árvores ou regras de decisão. De modo geral, essas abordagens possibilitam uma melhor compreensão do conhecimento contido no modelo do que os algoritmos baseados nos demais paradigmas de aprendizado (MICHALSKI; BRATKO; KUBAT, 1998; REZENDE, 2003; FACELI et al., 2011). Na literatura de séries temporais, abordagens baseadas em representações simbólicas têm sido adotadas para a construção de modelos de classificação inteligíveis (GEURTS, 2001; YAMADA et al., 2003; YE; KEOGH, 2011; HILLS et al., 2013). Nesse contexto, neste capítulo, são apresentados trabalhos da literatura relacionados à construção de classificadores e/ou a descritores de características que foram propostos para facilitar a interpretação do conhecimento representado pelos modelos.

#### 3.2 TRABALHOS RELACIONADOS

Um trabalho pioneiro para a construção de modelos de classificação simbólicos em séries temporais foi apresentado em Bakshi et al. (1994). Nesse estudo, os autores utilizaram formas morfológicas predefinidas como primitivas para descrever o comportamento das séries temporais. A partir dessas primitivas foram extraídas características relacionadas ao controle de processos químicos, as quais foram utilizadas para a indução de uma árvore de decisão.



Em [Kadous \(1999\)](#), foi proposto um método de classificação de séries temporais por meio de descritores que permitissem resultados compreensíveis, o qual foi posteriormente estendido em [Kadous e Sammut \(2004\)](#), [Kadous e Sammut \(2005\)](#). Nessa abordagem os autores realizaram a extração de descritores estatísticos globais, como a média dos valores da série temporal, e de características locais, como tendência e valores máximo e mínimo, por meio da divisão da série em subsequências. Posteriormente, foi aplicada uma técnica de agrupamento sobre as características extraídas, e então os centroides de cada grupo foram utilizados para a indução de uma árvore de decisão por meio do algoritmo C4.5.

No estudo elaborado em [Kudo, Toyama e Shimbo \(1999\)](#) os autores definiram regiões retangulares sobre o espaço-tempo das séries temporais para verificar quais séries possuíam valores dentro dessas regiões. Utilizando essa abordagem, as séries foram representadas por um conjunto de atributos binários que caracterizam a presença ou ausência da série temporal nas regiões definidas. A partir dessa representação, os autores criaram um algoritmo para a indução de regras de classificação.

Em [Rodriguez, Alonso e Bostrom \(2001\)](#), os autores propuseram dividir as séries em intervalos e então determinar descritores binários baseados nos valores de média e de variância. Essa representação foi utilizada para construção de regras baseadas na lógica de primeira ordem. Uma extensão desse trabalho foi apresentada em [Gonzalez e Diez \(2004\)](#), na qual características baseadas em distâncias foram adicionadas por meio da medida DTW. A partir dessa representação foram induzidos modelos de classificação baseados em árvores de decisão. Em [Morchen \(2003\)](#) os autores utilizaram os parâmetros da representação DWT e da DFT para a construção de regras de classificação.

No trabalho de [Cotofrei e Stoffel \(2002\)](#) e posteriormente em [Cotofrei e Stoffel \(2005\)](#), foram propostas duas abordagens para extração de características das séries temporais. A primeira baseia-se em particionar as séries em segmentos, para os quais são calculados valores de média e de variância, valor máximo e mínimo, e valor inicial e final de cada segmento. Na segunda abordagem, as séries são transformadas para a representação de diferenças de primeira ordem e então são discretizadas por meio do método de partição por frequências ([CATLETT, 1991](#)). Em seguida, a representação simbólica é também particionada em segmentos, para os quais regras de classificação são extraídas por meio do algoritmo C4.5. Como produto final, os autores propuseram combinar as regras de classificação induzidas sobre os descritores estatísticos com as regras extraídas sobre a representação simbólica.

Na abordagem proposta em [Yamada et al. \(2003\)](#), a classificação de séries temporais é realizada por meio da indução de uma árvore de decisão, em que cada nó interno está associado a uma série temporal. O processo de decisão é realizado pela

determinação do grau de similaridade entre uma nova série temporal a ser classificada e as séries representadas pelos nós internos da árvore.

Em [Hidasi e Gaspar-Papanek \(2011\)](#) foi apresentado o algoritmo *ShiftTree*, que baseia-se na construção de uma árvore de decisão, em que cada nó da estrutura está associado a um mecanismo dinâmico, denominado *cursor*, para determinar o valor dos atributos em cada série temporal avaliada. O *cursor* permite delimitar porções específicas das séries para cada tipo de descritor utilizado. Alguns dos descritores aplicados nesse trabalho foram média, variância, mínimo e máximo locais, mínimo e máximo globais.

No estudo conduzido por [Zalewski et al. \(2013\)](#), os autores propuseram primeiramente discretizar as séries temporais utilizando as representações SAX ([LIN et al., 2007](#)) ou EFVD ([ZALEWSKI et al., 2012b](#)). Após, com o intuito de caracterizar a relação temporal existente entre os símbolos produzidos pela discretização, os autores propuseram particionar as séries em segmentos de igual tamanho e então contabilizar a frequência dos símbolos e a frequência das transições entre os símbolos para cada segmento. Posteriormente, os descritores de cada segmento foram utilizados como atributos para a indução de uma árvore de decisão, por meio do algoritmo C4.5.

Apesar dos trabalhos mencionados terem como foco a capacidade de interpretação do conhecimento por meio da construção de modelos simbólicos de classificação, não necessariamente, os descritores utilizados permitem a fácil compreensão do conhecimento representado. Nesse contexto, alguns estudos têm defendido que características morfológicas podem ser utilizadas para fornecer uma maior capacidade de compreensão dos modelos simbólicos para a classificação de séries temporais.

No trabalho de [Geurts \(2001\)](#), os autores propuseram utilizar subsequências para caracterizar uma determinada classe. Nessa abordagem, cada nó interno da árvore de decisão é representado por uma subsequência identificada a partir de alguma série temporal do conjunto avaliado. O processo de classificação é realizado por meio do grau de similaridade entre uma dada série temporal e a subsequência presente em cada nó analisado.

Em [Maletzke et al. \(2009\)](#) os autores propuseram um método para a extração de conhecimento em séries temporais utilizando a combinação de descritores estatísticos, como valor máximo global, mínimo global, média, variância, curtose e obliquidade, e de descritores morfológicos, usando a identificação de *motifs* ([CHIU; KEOGH; LONARDI, 2003](#)). Por meio da avaliação experimental realizada os autores demonstraram a efetividade do método proposto, em termos da acurácia e da inteligibilidade do modelo de classificação, pela representação de padrões morfológicos em conjunção com árvores de decisão ([MALETZKE et al., 2014](#)).

Nos últimos cinco anos, o maior foco de esforço em pesquisas para a construção de modelos simbólicos para a classificação de séries temporais tem sido sobre a primitiva *shapelet*, a qual apresenta conceitos similares aos introduzidos em Geurts (2001), em Yamada et al. (2003) e em Maletzke et al. (2009).

### 3.3 PRIMITIVA SHAPELET

A primitiva *shapelet*, introduzida em Ye e Keogh (2009), tem sido explorada na literatura de séries temporais como um descritor de características morfológicas para prover um maior nível de inteligibilidade nos modelos simbólicos de classificação. Essa primitiva consiste em uma subsequência de uma série temporal que permite maximizar a discriminação entre as classes de um dado conjunto de séries temporais. A construção de classificadores por meio do uso de *shapelets* envolve a quantificação da similaridade entre uma *shapelet* e cada série temporal, de modo que o nível de similaridade é utilizado como característica discriminatória (LINES et al., 2012). A qualidade de cada *shapelet* é determinada em termos da capacidade de separação da classe à qual esta pertence em relação às demais classes do conjunto. Na figura 6 é apresentada uma ilustração de quatro séries temporais  $\{T_1, T_2, T_3, T_4\}$ , distribuídas em duas classes (A e B); e a *shapelet* identificada, que permite distinguir as séries da classe A.

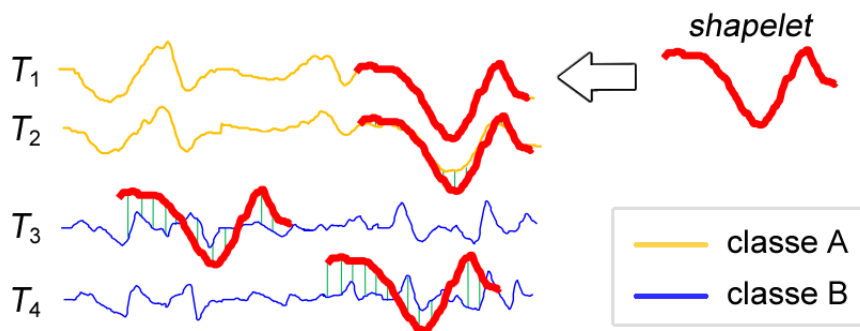


Figura 6 – Ilustração da primitiva *shapelet* para a classificação de séries temporais (YE; KEOGH, 2011).

A classificação de séries temporais por meio do uso da primitiva *shapelet* apresenta alguns benefícios em relação às demais abordagens, tais como:

- *shapelets* são facilmente interpretáveis e podem oferecer uma compreensão sobre o domínio do problema a ser tratado (YE; KEOGH, 2009; YE; KEOGH, 2011);
- os classificadores induzidos por meio de *shapelets* são mais compactos em relação a maioria das demais abordagens, e desse modo, o processo de classificação de novas instâncias torna-se mais rápido (MUEEN; KEOGH; YOUNG, 2011);

- o uso de *shapelets* permite a identificação de subsequências morfológicas similares, independente de fase, ou seja, não existe dependência em relação à posição temporal das subsequências na série (LINES et al., 2012; HILLS et al., 2013).

Os métodos de classificação baseados na primitiva *shapelet* podem ser divididos em duas categorias: a primeira corresponde à identificação de *shapelets* durante o processo de indução do classificador, aqui denominada por abordagem embutida (figura 7 (a)); enquanto que a segunda categoria é composta pelas abordagens que separam o processo de seleção de *shapelets* da etapa de construção do modelo de classificação, aqui denominada por abordagem transformada (figura 7 (b)).

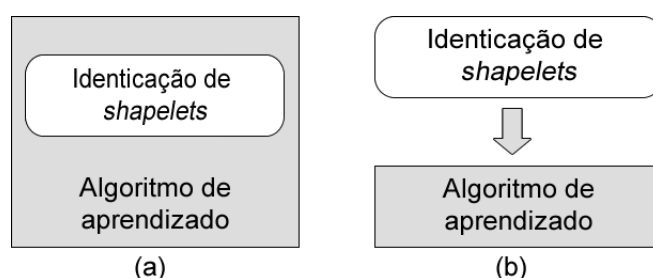


Figura 7 – Abordagens baseadas na primitiva *shapelet*. (a) Abordagem embutida e (b) abordagem transformada.

Nas seções 3.4 e 3.5 são apresentados exemplos de algoritmos propostos na literatura, considerando as abordagens embutida e transformada, respectivamente

### 3.4 ABORDAGEM EMBUTIDA – ALGORITMO YK

O conceito de *shapelet*, em sua essência, foi inicialmente proposto em Geurts (2001), no qual os autores estudaram a busca por subsequências em séries temporais que pudessem discriminar uma determinada classe. No entanto, os autores consideraram a busca pela melhor subsequência discriminante como intratável, devido a requisitos de tempo de processamento. Desse modo, o método alternativo utilizado nesse trabalho baseou-se em selecionar aleatoriamente uma série temporal de cada classe, e para cada série selecionada aplicar o algoritmo *adaptive piecewise constant approximation* (APCA) (CHAKRABARTI et al., 2002) para a redução de dimensão. Sobre esse espaço dimensional reduzido foi realizada a busca de subsequências de diferentes tamanhos com o intuito de selecionar uma subsequência com o menor nível de similaridade em relação às séries das demais classes.

Em Yamada et al. (2003), com o objetivo de construir um modelo interpretável, os autores propuseram um classificador baseado em uma árvore de decisão. No entanto, ao invés de realizar a extração de subsequências discriminantes para cada classe, como proposto em Geurts (2001), os autores utilizaram uma série temporal inteira

para representar cada nó da árvore de decisão induzida. Desse modo, o processo de classificação de novas instâncias é realizado pela mensuração do nível de similaridade entre a série temporal a ser classificada e as séries que compõem cada nó da árvore de decisão. Em relação ao trabalho de Geurts (2001), nessa abordagem não é utilizada redução de dimensão, mas mesmo assim, o custo computacional para encontrar uma série temporal que seja discriminante entre as demais classes é algumas ordens de grandeza menor do que a tarefa de busca por *shapelets*. Entretanto, como discutido em Ye e Keogh (2009), a utilização de características globais como séries temporais inteiras reduz significativamente a inteligibilidade do modelo de classificação.

Baseando-se nos trabalhos desenvolvidos em Geurts (2001) e em Yamada et al. (2003), o estudo realizado por Ye e Keogh (2009) propôs o algoritmo YK como resultado da combinação de ambas as abordagens desses trabalhos. Desse modo, ao invés de utilizar séries temporais inteiras para representar cada nó da árvore de decisão, como proposto em Yamada et al. (2003), o método faz uso da ideia introduzida por Geurts (2001), ao empregar somente as subsequências mais discriminantes das séries temporais de cada classe.

---

**Algoritmo 1:** SelecionarMelhorShapelet ( $B^M$ ,  $min$ ,  $max$ )

---

**Entrada:**  $B^M$ : conjunto de  $M$  séries temporais,  $min$ : tamanho mínimo de subsequência,  $max$ : tamanho máximo de subsequência

**Saída:** *shapelet*: subsequência de melhor qualidade identificada

```

1 shapelet  $\leftarrow \emptyset$ ;
2 melhor  $\leftarrow 0$ ;
3  $S_C \leftarrow \text{GerarSubsequenciasCandidatas}(B^M, min, max)$ ;
4 para cada subsequência  $S$  em  $S_C$  faça
5    $D_S \leftarrow \text{CalcularDistancias}(S, B^M)$ ;
6   qualidade  $\leftarrow \text{DeterminarQualidade}(D_S)$ ;
7   se qualidade > melhor então
8     melhor  $\leftarrow$  qualidade;
9     shapelet  $\leftarrow S$ ;
10  fim
11 fim
12 retorna shapelet

```

---

O processo de identificação de *shapelets*, segundo o algoritmo YK, é apresentado no algoritmo 1, caracterizado por três etapas: Gerar subsequências candidatas; calcular distâncias; e determinar qualidade.

### 3.4.1 Gerar subsequências candidatas

O algoritmo YK baseia-se na geração de todas as subsequências possíveis de um dado tamanho  $n$ , definido no intervalo  $[min, max]$ , onde  $min$  e  $max$  caracterizam o menor e o maior tamanho de subsequência possível, respectivamente. A geração de subsequências é realizada aplicando o conceito de janela deslizante, que é apresentado na definição 5.

**Definição 5 (Janela deslizante)** *Seja uma série temporal  $T$  de tamanho  $m$  e seja  $n$  o tamanho das subsequências possíveis, uma janela deslizante de tamanho  $n$  consiste em um conjunto formado por todas as subsequências distintas de tamanho  $n$  que podem ser extraídas de  $T$ .*

O processo de construção de uma janela deslizante baseia-se em selecionar uma subsequência de tamanho  $n$  a partir de cada posição  $p$  da série temporal  $T$ , denotada por  $S^{n,p} = \{t_p, \dots, t_{n+p-1}\}$ , tal que  $1 \leq p \leq (m - n + 1)$ . Como resultado, o conjunto de  $(m - n) + 1$  subsequências definido pela janela deslizante pode ser expresso por:

$$S_T^n = \{S^{n,1} \cup S^{n,2} \cup \dots \cup S^{n,(m-n+1)}\} \quad (3.1)$$

Na figura 8 é ilustrado o processo de aplicação de uma janela deslizante de tamanho  $n$  sobre uma dada série temporal  $T$ . Na porção inicial dessa figura são representadas oito subsequências de tamanho  $n$  extraídas das posições  $p = \{1, \dots, 8\}$ ; enquanto que na porção central é representada a subsequência de tamanho  $n$  extraída a partir da posição  $p = 67$ .

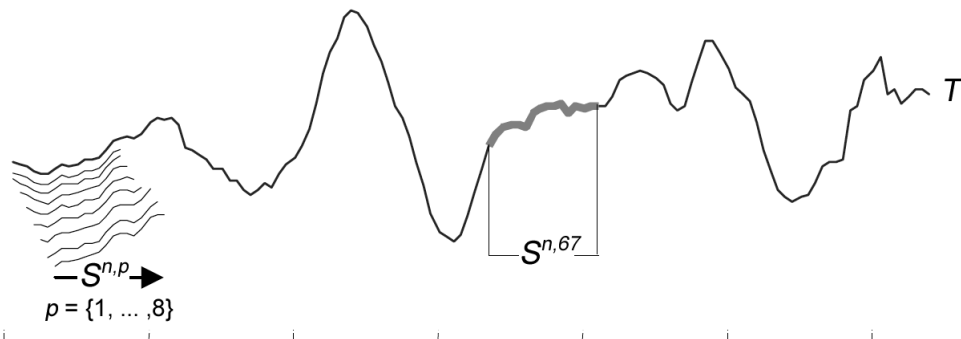


Figura 8 – Ilustração do processo de aplicação de uma janela deslizante (KEOGH; LIN; TRUPPEL, 2003).

Seja uma base de treino  $B^M$  com  $M$  séries temporais, denota-se o conjunto de todas as subsequências candidatas de tamanho  $n$  para uma série temporal  $T_i \in$

$B^M$  como  $S_i^n$ , tal que  $1 \leq i \leq M$ . Desse modo, denota-se o conjunto de todas as subsequências possíveis de tamanho  $n$  em  $B^M$  como:

$$S^n = \{S_1^n \cup S_2^n \cup \dots \cup S_M^n\} \quad (3.2)$$

e a quantidade de subsequências candidatas contidas em  $S^n$  é dada por:

$$|S^n| = \sum_{T_i \in B^M} (m_i - n + 1) \quad (3.3)$$

onde  $m_i$  é o tamanho da série temporal  $T_i$ .

Assumindo que para uma determinada base  $B^M$  o tamanho  $n$  seja restrito ao intervalo  $\min \leq n \leq \max$ , denota-se o conjunto de todas as subsequências candidatas como:

$$S_C = \{S^{\min} \cup S^{\min+1} \cup \dots \cup S^{\max}\} \quad (3.4)$$

e o número total de elementos do conjunto  $S_C$  dado por:

$$|S_C| = \sum_{n=\min}^{\max} \sum_{T_i \in B^M} (m_i - n + 1) \quad (3.5)$$

---

**Algoritmo 2:** GerarSubsequênciasCandidatas ( $B^M$ ,  $\min$ ,  $\max$ )

---

**Entrada:**  $B^M$ : conjunto de séries  $M$  temporais,  $\min$ : tamanho mínimo de subsequência,  $\max$ : tamanho máximo de subsequência

**Saída:**  $S_C$ : conjunto de subsequências candidatas

```

1  $S_C \leftarrow \emptyset$ ;
2  $salto \leftarrow 1$ ;
3  $n \leftarrow \max$ ;
4 enquanto  $n \geq \min$  faça
5   para cada série temporal  $T$  em  $B^M$  faça
6      $S_C \leftarrow S_C \cup S_T^n$ ;
7   fim
8    $n \leftarrow n - salto$ ;
9 fim
10 retorna  $S_C$ 
```

---

No algoritmo 2 é descrito o processo de geração das subsequências candidatas. A variável *salto* é utilizada para determinar a variação do tamanho de subsequência a

cada iteração. Em [Ye e Keogh \(2009\)](#), [Ye e Keogh \(2011\)](#) na maioria das avaliações experimentais realizadas, foi adotado  $\text{salto} = 1$ .

Em relação aos parâmetros  $\text{min}$  e  $\text{max}$  os autores do algoritmo YK propuseram utilizar, na maioria de seus experimentos, os seguintes valores:  $\text{min} = 3$ , que define o menor tamanho de subsequência significante; e  $\text{max} = m$ .

### 3.4.2 Calcular distâncias

A segunda etapa do algoritmo YK consiste em calcular a distância de uma dada subsequência candidata  $S_i^n \in S_C$  para cada série temporal  $T \in B^M$ , tal como descrito no algoritmo 3.

---

**Algoritmo 3:** CalcularDistâncias ( $S, B^M$ )

---

**Entrada:**  $B^M$ : conjunto de séries  $M$  temporais,  $S$ : subsequência de tamanho  $n$

**Saída:**  $D_S$ : conjunto de distâncias de  $S$  em relação a  $B^M$

---

```

1  $D_S \leftarrow \emptyset$ ;
2 para cada série temporal  $T$  em  $B^M$  faça
3    $D_S \leftarrow D_S \cup \text{SubDist}(S, T)$ ;
4 fim
5 retorna  $D_S$ 
```

---

A função *SubDist*, utilizada na linha 3, é caracterizada pela definição 7. No entanto, anteriormente, é necessário apresentar a distância entre séries temporais (definição 6).

**Definição 6 (Distância entre séries temporais)** *Sejam duas séries temporais  $T_a$  e  $T_b$ , de igual tamanho  $m$ , a função  $\text{Dist}(T_a, T_b)$  retorna um valor real não negativo  $d$ , que expressa a distância entre as duas séries temporais.*

A função *Dist* também pode ser utilizada para calcular a distância entre duas subsequências de mesmo tamanho. No entanto, o processo de identificação de *shapels* envolve o cálculo de similaridade entre subsequências de tamanhos diferentes. Desse modo, tem-se a definição do conceito de distância de subsequência.

**Definição 7 (Distância de subsequência)** *Sejam duas subsequências  $S$  e  $R$  de tamanho  $n_S$  e  $n_R$ , respectivamente, tal que  $n_S < n_R$ , e seja  $S_R^{n_S}$  a janela deslizante de tamanho  $n_S$  aplicada sobre a subsequência  $R$ , a distância de subsequência é dada pela função  $\text{SubDist}(S, R)$ :*

$$\text{SubDist}(S, R) = \text{Min}(\text{Dist}(S, r)), \quad r \in S_R^{n_S} \quad (3.6)$$



onde  $Min$  é uma função que retorna a menor distância entre  $S$  e todas as subsequências contidas em  $S_R^{ns}$ .

Na figura 9 é apresentada uma ilustração da posição na qual ocorre a menor distância entre as subsequências  $S$  e  $R$ .

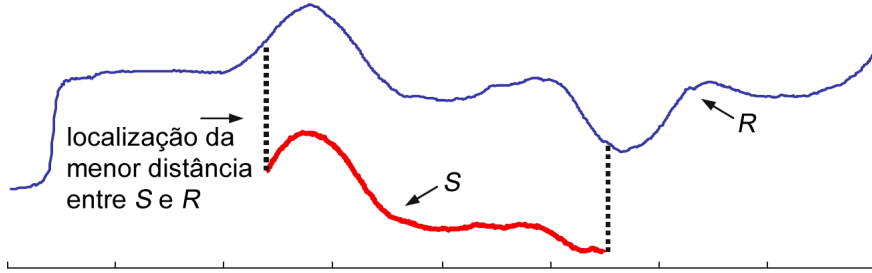


Figura 9 – Representação esquemática da localização da menor distância encontrada entre as subsequências  $S$  e  $R$  (YE; KEOGH, 2011).

Para o algoritmo YK, os autores propuseram definir a função  $Dist$  por meio da distância Euclidiana (equação 2.5) e aplicar a normalização de amplitude (equação 2.2) sobre as subsequências, anteriormente ao cálculo de distâncias.

Como produto dessa etapa, tem-se um conjunto de distâncias (definição 8) que permite caracterizar uma determinada subsequência em relação a um conjunto de séries temporais.

**Definição 8 (Conjunto de distâncias)** *Seja uma subsequência  $S$  de tamanho  $n$  e uma base de séries temporais  $B^M$ , composta por séries temporais de tamanho  $m$ , tal que  $n \leq m$ , o conjunto de distâncias de  $S$  é expresso por:*

$$D_S = \{d_1, d_2, \dots, d_M\} = \{SubDist(S, T_1), SubDist(S, T_2), \dots, SubDist(S, T_M)\} \quad (3.7)$$

### 3.4.3 Determinar qualidade

A terceira etapa do algoritmo YK para a identificação de *shapelets*, descrita pelo algoritmo 4, consiste em determinar a qualidade de uma subsequência  $S$  a partir do respectivo conjunto de distâncias  $D_S$ , calculadas durante a etapa anterior. Para isso, inicialmente, conjunto de distâncias  $D_S$  é transformado em uma linha de ordenação (definição 9).

**Definição 9 (Linha de ordenação)** *Seja um conjunto de distâncias  $D_S$ , e seja uma base de séries temporais  $B^M$ , com  $M$  séries temporais e um conjunto  $C$  de  $w$  classes. Uma linha de ordenação  $L$  é uma representação unidimensional de  $D_S$  por meio de um conjunto de  $M$  valores, tal que  $L = \{d_1, d_2, \dots, d_M\}$  esteja ordenado de modo ascendente em função dos valores  $d_i$ .*

---

**Algoritmo 4:** DeterminarQualidade( $D_S$ )
 

---

**Entrada:**  $D_S$ : conjunto de distâncias de  $S$  em relação a  $B^M$

**Saída:**  $maxGl$ : máximo valor de ganho de informação

```

1  $L \leftarrow Ordenar(D_S)$ ;
2  $maxGl \leftarrow 0$ ;
3  $maxSep \leftarrow 0$ ;
4 para  $i \leftarrow 1$  até  $|L| - 1$  faça
5    $ls_i \leftarrow \frac{1}{2}(d_{i+1} + d_i)$ ;
6    $Gl \leftarrow Ganho(L, ls_i)$ ;
7    $Sep \leftarrow DistSep(L, ls_i)$ ;
8   se ( $Gl > maxGl$ ) ou (( $Gl = maxGl$ ) e ( $Sep > maxSep$ )) então
9      $maxGl \leftarrow Gl$ ;
10     $maxSep \leftarrow Sep$ ;
11  fim
12 fim
13 retorna  $maxGl$ 
  
```

---

Na figura 10 é apresentada uma representação esquemática da construção de uma linha de ordenação para uma dada subsequência  $S$  em relação a uma base de séries temporais  $B^8 = \{T_1, \dots, T_8\}$ , composta por séries da classe círculo e quadrado.

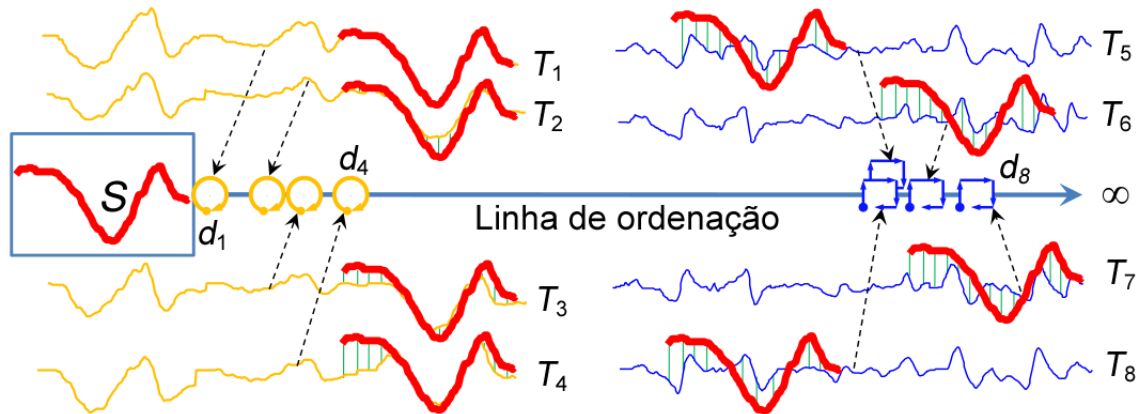


Figura 10 – Representação esquemática de uma linha de ordenação (MUEEN; KEOGH; YOUNG, 2011).

Na linha 5 do algoritmo 4, cada par adjacente dos elementos contidos na linha de ordenação  $L$ , é utilizado para determinar o valor do limiar de separação, tal como apresentado na definição 10.

**Definição 10 (Limiar de separação)** Um limiar de separação  $ls$  é um número real que permite dividir uma linha de ordenação  $L$  em dois subconjuntos disjuntos  $L_E = \{d_i : d_i \in L, d_i \leq ls\}$  e  $L_D = \{d_i : d_i \in L, d_i > ls\}$ .

Seja uma linha de ordenação  $L$ , o conjunto de limiares de separação possíveis é dado por  $LS = \{ls_1, ls_2, \dots, ls_{M-1}\}$ , tal que:

$$ls_i = \frac{d_{i+1} + d_i}{2} \quad (3.8)$$

A figura 11 contém uma ilustração de uma linha de ordenação composta por nove séries temporais, sendo cinco da classe círculo e quatro da classe quadrado. Nesta figura, as posições indicadas por  $ls_4$  e  $ls_5$  representam o limiar de separação entre as séries  $L_4$  e  $L_5$ , e  $L_5$  e  $L_6$ , respectivamente.

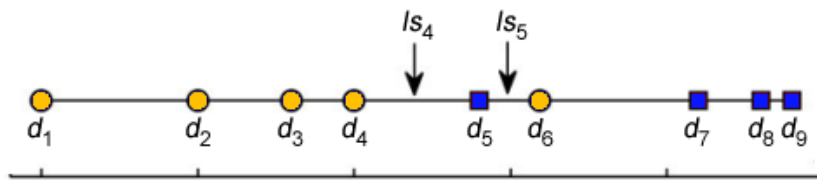


Figura 11 – Ilustração do limiar de separação (GHALWASH; OBRADOVIC, 2012).

Na proposta do algoritmo YK, o ganho de informação (SHANNON, 1948) é utilizado como medida de qualidade do limiar de separação. Essa medida é determinada pela função *Ganho* (linha 6), caracterizada nas definições 11 e 12.

**Definição 11 (Ganho de informação)** Considerando um determinado limiar de separação  $ls$ , e seja o tamanho de  $L$ ,  $L_E$  e  $L_D$ , expresso por  $|L|$ ,  $|L_E|$  e  $|L_D|$  respectivamente; o ganho de informação é dado por:

$$Ganho(L, ls) = Entropia(L) - \left( \frac{|L_E|}{|L|} Entropia(L_E) + \frac{|L_D|}{|L|} Entropia(L_D) \right) \quad (3.9)$$

**Definição 12 (Entropia)** Seja uma linha de ordenação  $L$ , e seja  $n_k$  a quantidade de elementos de  $L$  que estejam associados a uma determinada classe  $c_k \in C$ , a entropia de  $L$  é expressa por:

$$Entropia(L) = \sum_{k=1}^w -\frac{n_k}{N} \log \frac{n_k}{N} \quad (3.10)$$

no qual  $N = \sum_{j=1}^w n_j$ .

Na abordagem proposta para o algoritmo YK, distintos limiares de separação de uma mesma subsequência podem estar associados a um mesmo valor de ganho de informação. Nesse cenário os autores propuseram a adoção da distância de separação como um critério de desempate.

**Definição 13 (Distância de separação)** Considerando um determinado limiar de separação  $ls$ , a distância de separação é dada por:

$$DistSep(L, ls) = \frac{1}{|L_D|} \sum_{d_i \in L_D} d_i - \frac{1}{|L_E|} \sum_{d_i \in L_E} d_i \quad (3.11)$$

No exemplo ilustrativo da figura 11, os limiares de separação  $ls_4$  e  $ls_5$  são os maiores valores de ganho de informação encontrados para a linha de ordenação representada, e ambas têm o mesmo valor, pois separam igualmente as séries entre as classes. Nesse caso, o limiar de separação  $ls_4$  será selecionado pois apresenta a maior distância de separação.

Dadas as definições apresentadas acima, os autores de Ye e Keogh (2009), definem uma *shapelet* como:

**Definição 14 (Shapelet)** Uma *shapelet* é uma tupla  $S_H = (S, ls)$ , na qual  $S$  é alguma subsequência de  $S_C$ , que associada a um valor  $ls$  permite dividir uma linha de ordenação  $L$  em dois subconjuntos disjuntos, tal que o ganho de informação gerado por  $ls$  seja o máximo possível para  $L$ .

Utilizando esta definição, nas seções 3.4.4 e 3.4.5 são descritos os processos de indução da árvore de decisão e de classificação, respectivamente.

#### 3.4.4 Construção da árvore de decisão

O processo de identificação de *shapelets*, realizado pelo algoritmo YK, ocorre conjuntamente com a indução da árvore de decisão. No algoritmo 5 é descrito um procedimento recursivo, que recebe como parâmetros de entrada um conjunto de séries temporais  $B^M$ , uma árvore de decisão  $A$ , inicialmente vazia, e os valores de *min* e de *max*, previamente determinados. Em cada chamada do algoritmo é realizada a busca pela melhor *shapelet*  $S_H$  para um dado conjunto de séries temporais  $B^M$ . Desse modo, para cada  $S_H$  selecionada, um novo nó de decisão é criado em  $A$ .

Na figura 12 é apresentada uma representação esquemática do processo de construção de uma árvore de decisão por meio de *shapelets*. A figura 12 (a) contém a ilustração de sete séries temporais, cada uma relacionada a uma classe de  $\{0, 1, 2, 3, 4, 5, 6\}$ . Na figura 12 (b) é apresentada a árvore de decisão construída, na qual cada nó é representado por uma das *shapelets* ilustradas na figura 12 (c).

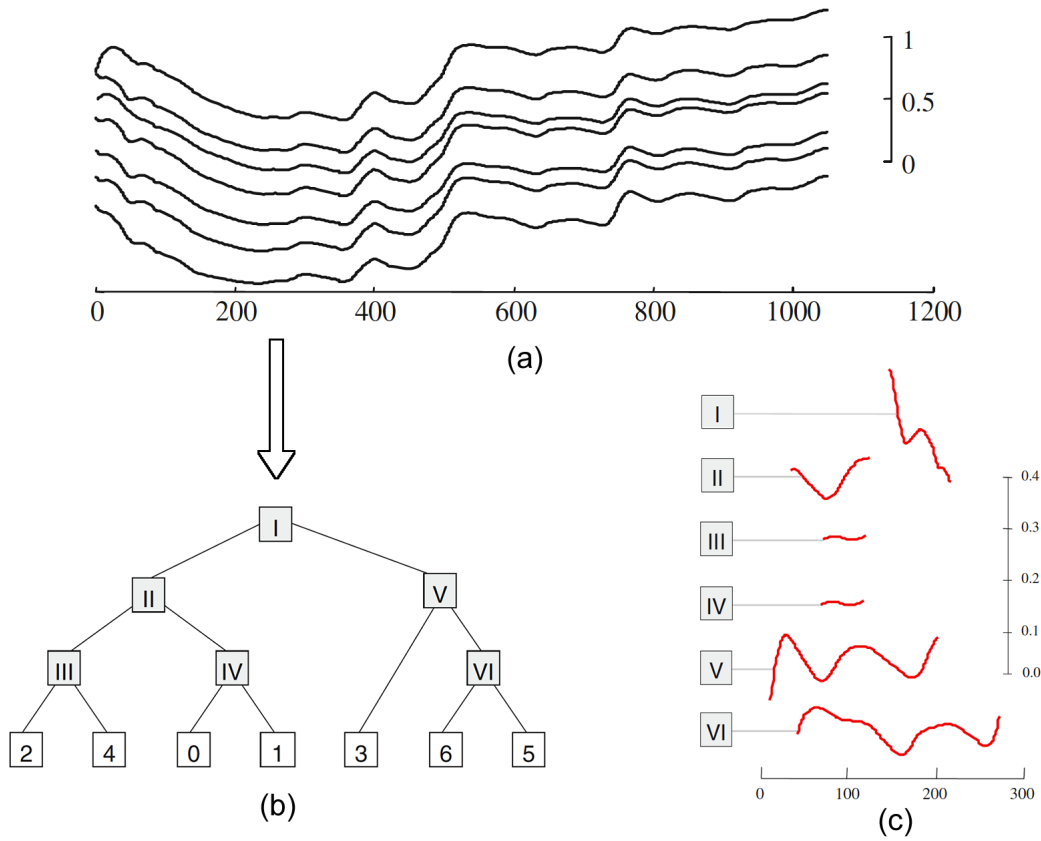


Figura 12 – Representação esquemática da construção de uma árvore de decisão construída com *shapelets*. (a) sete séries temporais, cada qual representando uma classe; (b) árvore de decisão gerada com as (c) *shapelets* identificadas (YE; KEOGH, 2011).

---

**Algoritmo 5:** AlgoritmoYK ( $B^M$ ,  $min$ ,  $max$ ,  $A$ )

---

**Entrada:**  $B^M$ : conjunto de  $M$  séries temporais,  $min$ : tamanho mínimo de subsequência,  $max$ : tamanho máximo de subsequência,  $A$ : árvore de decisão vazia

**Saída:**  $A$ : árvore de decisão construída

```

1  $S_H \leftarrow \text{SelecionarMelhorShapelet}(B^M, min, max);$ 
2  $\text{CriarNodo}(A, S_H);$ 
3 se  $\text{Entropia}(L_E) \neq 0$  então
4   |  $\text{AlgoritmoYK}(B^{L_E}, min, max, A);$ 
5 fim
6 se  $\text{Entropia}(L_D) \neq 0$  então
7   |  $\text{AlgoritmoYK}(B^{L_D}, min, max, A);$ 
8 fim
```

---

Como mencionado nas subseções anteriores, uma *shapelet*  $S_H$  permite dividir uma linha de ordenação  $L$  em duas partições  $L_E$  e  $L_D$ . Consequentemente, também é

possível separar uma dada base de séries temporais  $B^M$  em duas partições  $B^{L_E}$  e  $B^{L_D}$ . Desse modo, em cada chamada do algoritmo YK (algoritmo 5), duas novas chamadas recursivas são realizadas, uma para considerar somente as séries temporais contidas em  $B^{L_E}$  e a outra para as demais contidas em  $B^{L_D}$ . A construção da árvore de decisão  $A$  se encerra quando a entropia das partições esquerda e direita forem iguais à zero.

### 3.4.5 Classificação

Após a construção da árvore de decisão  $A$ , o processo de classificação de uma nova instância  $T$  é realizado por meio de sucessivas comparações das distâncias de subsequência entre a série temporal  $T$  e a *shapelet*  $S_H$  contida em cada nó analisado, até que um nó folha seja encontrado. No algoritmo 6 é apresentado um procedimento recursivo para prever a classe de uma série temporal  $T$ , segundo uma dada árvore de decisão  $A$ . A tarefa de classificação inicia-se por verificar a distância de subsequência entre  $T$  e a *shapelet*  $S_H$  que está na raiz da árvore  $A$ . Caso a distância calculada seja menor que o limiar de separação de  $S_H$ , o próximo nó a ser verificado estará na sub-árvore esquerda de  $A$ , caso contrário estará na sub-árvore direita de  $A$ .

---

#### Algoritmo 6: PredizerClasse ( $A, T$ )

---

**Entrada:**  $A$ : árvore de decisão,  $T$ : série temporal a ser classificada

**Saída:** rótulo da classe predita

```

1 se  $A$  é uma folha então
2   | retorna classe de  $A$ 
3 fim
4 senão
5   |  $S_H \leftarrow$  shapelet da raiz de  $A$ ;
6   | se  $SubDist(S_H(S), T) < S_H(l_s)$  então
7   |   |  $PredizerClasse(subarvore\ esquerda\ de\ A, T)$ ;
8   |   | fim
9   |   | senão
10  |   |   |  $PredizerClasse(subarvore\ direita\ de\ A, T)$ ;
11  |   |   | fim
12 fim
```

---

### 3.4.6 Escalabilidade

O processo de identificação de *shapelets* pode ser entendido como uma versão supervisionada do algoritmo de descoberta de *motifs* (BUHLER; TOMPA, 2002; CHIU; KEOGH; LONARDI, 2003). Desse modo, a seleção de *shapelets* consiste em um procedimento de busca exaustiva sobre todo o conjunto de treinamento. O custo com-

putacional de tempo do algoritmo YK é  $O(M^2m^4)$  no pior caso, onde  $M$  corresponde ao número de séries temporais do conjunto de treinamento e  $m$  ao tamanho da maior série do conjunto. Em Ye e Keogh (2009), Ye e Keogh (2011), os autores propuseram utilizar algumas técnicas para minimizar o impacto desse custo, no entanto sem melhorias em relação ao pior caso.

### 3.4.7 Logical shapelets

No trabalho de Mueen, Keogh e Young (2011) foi proposta uma extensão do método de Ye e Keogh (2011), Ye e Keogh (2009), na qual os autores introduziram o algoritmo *logical shapelets*. Os autores argumentam que existem séries temporais que somente podem ser descritas, ou melhor caracterizadas, por meio da combinação de *shapelets*. Desse modo, nessa abordagem foram utilizados múltiplos *shapelets*, agrupados por meio de conjunções ou disjunções, em cada nó da árvore de decisão. A principal motivação desse algoritmo é exemplificada na figura 13. No cenário representado, existem duas classes, classe quadrado (figura 13 (a)) e classe círculo (figura 13 (b)), distribuídas em quatro séries temporais. Nenhuma das *shapelets* selecionadas,  $S_H^1$  e  $S_H^2$ , são capazes de prover uma boa separação das séries na linha de ordenação (figura 13 (c)), pois estão presentes em ambas as classes. Desse modo, a proposta do algoritmo *logical shapelets*, que consiste em combinar *shapelets* por meio de conjunções ou disjunções, pode resolver esse tipo de problema.

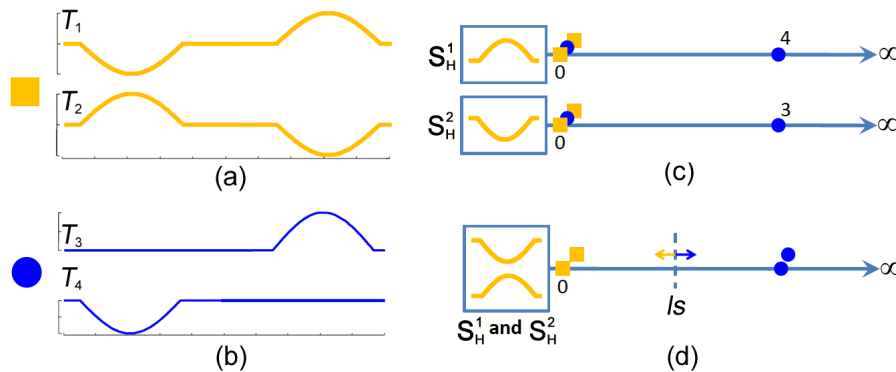


Figura 13 – Ilustração da ideia do algoritmo *logical shapelets*. (a) séries da classe quadrado; (b) séries da classe círculo; (c) linhas de ordenação geradas pelas *shapelets* selecionadas; (d) linha de ordenação produzida pela conjunção das *shapelets*  $S_H^1$  e  $S_H^2$  (MUEEN; KEOGH; YOUNG, 2011).

Os autores conduziram uma avaliação experimental com alguns estudos de caso e demonstraram a efetividade da abordagem proposta em termos da acurácia do modelo classificação. Os resultados do algoritmo *logical shapelets* foram comparados com o algoritmo YK e com o algoritmo 1-vizinho mais próximo usando distância Euclidiana e *Dynamic Time Warping* (DTW) (descritos na seção 2.7). Além desse aspecto, nesse trabalho foram propostas algumas técnicas de otimização para a busca

de *shapelets*, dentre as quais, uma abordagem baseada em armazenar cálculos de distâncias recorrentes possibilitou a redução da complexidade de tempo do algoritmo para  $O(M^2m^3)$ , no pior caso.

### 3.4.8 Fast shapelets

No estudo de [Keogh e Rakthanmanon \(2013\)](#) foi apresentado um novo método baseado no algoritmo YK, denominado *fast shapelets*, com o principal intuito de reduzir o custo computacional do processo de identificação de *shapelets*. Nesse estudo, os autores propuseram, primeiramente, discretizar as séries temporais utilizando o algoritmo *symbolic aggregate approximation* (SAX) e após, utilizar o método de projeções aleatórias ([BUHLER; TOMPA, 2002](#)) para determinar o grau de similaridade entre as séries discretizadas e também a qualidade das subsequências. Por meio dessa representação e da introdução de uma técnica de otimização baseada na normalização das subsequências durante o processo de cálculo de distâncias, o algoritmo *fast shapelets* possibilitou a redução da complexidade de tempo para  $O(Mm^2)$ , no pior caso. Por outro lado, a abordagem adotada baseia-se em um método não determinístico (projeções aleatórias), de modo que a cada execução do algoritmo, um subconjunto distinto de *shapelets* pode ser identificado. Além desse aspecto, o algoritmo *fast shapelets* introduziu no processo de busca pelas *shapelets* quatro novos parâmetros, que necessitam ser avaliados para cada domínio.

Os autores conduziram uma avaliação experimental utilizando 32 conjuntos de séries temporais amplamente utilizados na literatura ([KEOGH et al., 2006](#)) e demonstraram que não existe diferença significativa, em termos da acurácia do modelo de classificação, entre os algoritmos *logical shapelets* e *fast shapelets*.

## 3.5 ABORDAGEM TRANSFORMADA

Em [Lines et al. \(2012\)](#), os autores introduziram uma generalização da primitiva *shapelet* para separar a tarefa de identificação de *shapelets* do processo de indução da árvore de decisão. Essa abordagem consiste em extrair as  $k$  melhores *shapelets* do conjunto de treinamento para construir uma nova representação das séries temporais. Os autores denominaram essa representação como transformada *shapelet*, na qual cada série temporal passa a ser representada pelo conjunto de distâncias em relação ao conjunto de  $k$  *shapelets* selecionado.

O processo de aplicação da transformada *shapelet* pode ser estruturado em três fases:

1. Estimar o parâmetro  $k$ ;



2. Selecionar  $k$  *shapelets*;
3. Aplicar a transformada.

### 3.5.1 Estimar o parâmetro $k$

A fase 1 da transformada *shapelet* consiste em estimar automaticamente a quantidade  $k$  de *shapelets* que serão utilizadas. O parâmetro  $k$  é utilizado somente como um limiar de corte, de modo que o valor de  $k$  não tem influência sobre a qualidade individual das *shapelets* selecionadas. Entretanto, a seleção de poucas *shapelets* pode não prover informações suficientes para determinados domínios, enquanto que o uso de muitas *shapelets* pode causar sobreajuste nos modelos construídos sobre o conjunto de treinamento. Em [Lines et al. \(2012\)](#) foram propostas duas abordagens para estimar o valor de  $k$ . Na primeira, o valor de  $k$  é ajustado para  $\frac{m}{2}$ , sendo  $m$  o tamanho das séries temporais. A segunda abordagem baseia-se na aplicação de um processo de validação cruzada de cinco partições sobre o conjunto de treinamento. Para cada partição foram criados  $m$  conjuntos, cada qual com  $k \in \{1, 2, \dots, m\}$  *shapelets*. O valor final de  $k$  é definido pelo valor de  $m$  que possibilitou a maior acurácia média sobre as cinco partições.

### 3.5.2 Selecionar $k$ *shapelets*

A fase 2 é descrita pelo algoritmo 7, o qual apresenta algumas similaridades em relação ao processo de escolha de *shapelets* adotado pelo algoritmo YK. No entanto, as propostas de [Ye e Keogh \(2009\)](#) e de [Lines et al. \(2012\)](#) para a identificação de *shapelets* se distinguem principalmente em dois aspectos: a generalização da medida de qualidade, e a manutenção de  $k$  *shapelets* durante o processo de busca.

Os autores de [Lines et al. \(2012\)](#) introduziram uma generalização do processo de determinação da qualidade de uma *shapelet*, de modo que a medida utilizada não necessariamente tenha que estar associada a um limiar de separação. Para avaliar essa nova ideia os autores propuseram aplicar a estatística  $F$ , do cálculo de variâncias, para avaliar a qualidade da distribuição das classes em um dado conjunto de distâncias  $D_S$ . No entanto, cabe ressaltar que a abordagem continua sendo válida também para o ganho de informação como medida de qualidade.

Considerando a generalização proposta, é possível redefinir o conceito da primitiva *shapelet* como:

**Definição 15 (*Shapelet*)** Uma *shapelet*  $S_H$  é alguma subsequência  $S \in S_C$ , que por meio de alguma métrica de qualidade, permite uma melhor distribuição das séries temporais de um conjunto  $B^M$  para um dado conjunto  $C$  de classes.

---

**Algoritmo 7:**  $k$ MelhoresShapelets ( $B^M$ ,  $min$ ,  $max$ ,  $k$ )
 

---

**Entrada:**  $B^M$ : conjunto de  $M$  séries temporais,  $min$ : tamanho mínimo de subsequência,  $max$ : tamanho máximo de subsequência,  $k$ : quantidade de *shapelets* a serem selecionadas

**Saída:**  $kShapelets$ : conjunto de  $k$  melhores *shapelets* identificadas

```

1  $kShapelets \leftarrow \emptyset$ ;
2  $shapelets' \leftarrow \emptyset$ ;
3  $S_C \leftarrow GerarSubsequenciasCandidatas(B^M, min, max)$ ;
4 para cada subsequência  $S$  em  $S_C$  faça
5    $D_S \leftarrow CalcularDistancias(S, B^M)$ ;
6    $qualidade \leftarrow DeterminarQualidade(D_S)$ ;
7    $shapelets' \leftarrow shapelets' \cup (S, qualidade)$ ;
8 fim
9  $OrdenarPorQualidade(shapelets')$ ;
10  $RemoverAutoSimilares(shapelets')$ ;
11  $kShapelets \leftarrow SelecionarKMelhores(k, kShapelets \cup shapelets')$ ;
12 retorna  $kShapelets$ 

```

---

Diferente da proposta de [Ye e Keogh \(2009\)](#), na qual somente a *shapelet* de melhor qualidade é mantida durante o processo de busca, na abordagem de [Lines et al. \(2012\)](#) são mantidas as  $k$  melhores. Para isso foram utilizados três procedimentos (linhas 10-12). A cada iteração do algoritmo, todas as subsequências de tamanho  $n$  são armazenadas na variável *shapelets'*; ao final de cada iteração, as subsequências são organizadas por meio do procedimento *OrdenarPorQualidade*, de modo ascendente em função da medida de qualidade utilizada. Após, o procedimento *RemoverAutoSimilares* é aplicado para retirar do conjunto *shapelets'* as *shapelets* consideradas auto-similares. Os autores definem a auto-similaridade de duas *shapelets* quando ambas são provenientes de uma mesma série temporal e apresentam interseção entre seus índices de início e fim, dentro da série. Ao final, por meio da função *SelecionarKMelhores* é realizado um filtro das  $k$  melhores *shapelets* a partir da união dos conjuntos  $kShapelets$  e *shapelets'*.

Com o intuito de minimizar o custo computacional do processo de identificação de *shapelets*, em [Lines et al. \(2012\)](#) os autores propuseram uma abordagem automática para determinar os parâmetros  $min$  e  $max$ , descrita no algoritmo 8. O algoritmo *EstimarMinMax* não tem o intuito de garantir necessariamente a escolha dos melhores valores para os parâmetros  $min$  e  $max$ , e sim permitir um procedimento automático para estimá-los. Para cada iteração do algoritmo, a ideia consiste em selecionar aleatoriamente dez séries temporais de  $B^M$ , que serão utilizadas na etapa de treinamento (linhas 2-3); após, o algoritmo 7 é aplicado para selecionar  $k = 10$  *shapelets*, considerando

os parâmetros:  $B^{10}$  que representa as dez séries temporais selecionadas,  $min = 3$  e  $max = M$ . Ao executar as 10 iterações do algoritmo, o conjunto  $shapelets^*$  contém 100 *shapelets* ordenadas com base no tamanho das *shapelets* (linha 7). Na proposta de Lines et al. (2012), os parâmetros  $min$  e  $max$  são definidos pelo tamanho das *shapelets* situadas nas posições 25 e 75 do conjunto  $shapelets^*$ , respectivamente.

---

**Algoritmo 8:** EstimarMinMax ( $B^M$ )

---

**Entrada:**  $B^M$ : conjunto de  $M$  séries temporais

**Saída:**  $min$ : tamanho mínimo de subsequência,  $max$ : tamanho máximo de subsequência

```

1  $shapelets^* \leftarrow \emptyset$ ;
2 para  $i \leftarrow 1$  até 10 faça
3    $OrganizarAleatoriamente(B^M)$ ;
4    $B^{10} \leftarrow \{T_1, T_2, \dots, T_{10}\}$ ;
5    $shapelets^* \leftarrow shapelets^* \cup kMelhoresShapelets(B^{10}, 3, M, 10)$ ;
6 fim
7  $OrdenarPeloTamanhoDeSubsequencia(shapelets^*)$ ;
8  $min \leftarrow tamanho(S_H^{25})$ ;
9  $max \leftarrow tamanho(S_H^{75})$ ;
10 retorna ( $min, max$ )

```

---

### 3.5.3 Aplicar a transformada

Na fase 3, o conjunto  $kShapelets$  gerado na fase anterior é utilizado para transformar um conjunto de séries temporais  $B^M$  de  $w$  classes em uma representação atributo-valor (tabela 1). Nessa representação cada série temporal  $T_i \in B^M$ ,  $1 \leq i \leq M$ , é descrita por um conjunto de  $k$  *shapelets*, cada qual associada a um valor distância  $d_{ij} = SubDist(S_H^j, T_i)$ ,  $1 \leq j \leq k$ .

Tabela 1 – Representação atributo-valor da transformada *shapelet*.

Séries Temporais ( $T$ )	$S_H^1$	$S_H^2$	...	$S_H^k$	Classe ( $c$ )
$T_1$	$d_{11}$	$d_{12}$	...	$d_{1k}$	$c_1$
$T_2$	$d_{21}$	$d_{22}$	...	$d_{2k}$	$c_2$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$T_M$	$d_{M1}$	$d_{M2}$	...	$d_{Mk}$	$c_w$

O algoritmo 9 representa a descrição do processo da transformada *shapelet*, no qual a função *TransformadaShapelet* recebe como entrada o conjunto de séries temporais  $B^M$  a ser transformado e o conjunto  $kShapelets$ . A cada iteração do algoritmo uma série temporal  $T_i$  é selecionada, e a distância de subsequência  $d_{ij}$  é calculada

para cada *shapelet*  $S_H^j$ . Para cada série  $T_i$  as distâncias são armazenadas em  $D_{T_i} = \{d_{i1}, d_{i2}, \dots, d_{ik}\}$ ; e ao final, a função retorna o conjunto  $X^{\mathbb{R}} = \{D_{T_1} \cup D_{T_2} \cup \dots \cup D_{T_M}\}$ , que é a representação atributo-valor da transformada *shapelet*.

---

**Algoritmo 9:** TransformadaShapelet ( $B^M, kShapelets$ )

---

**Entrada:**  $B^M$ : conjunto de séries  $M$  temporais,  $kShapelets$ : conjunto de  $k$  melhores *shapelets* identificadas

**Saída:**  $X^{\mathbb{R}}$ : representação atributo-valor da transformada *shapelet*

```

1  $X^{\mathbb{R}} \leftarrow \emptyset$ ;
2 para cada série temporal  $T_i$  em  $B^M$  faça
3   para cada shapelet  $S_H^j$  em  $kShapelets$  faça
4      $d_{ij} \leftarrow \text{SubDist}(S_H^j, T_i)$ ;
5      $D_{T_i} \leftarrow D_{T_i} \cup d_{ij}$ ;
6   fim
7    $X^{\mathbb{R}} \leftarrow X^{\mathbb{R}} \cup D_{T_i}$ ;
8 fim
9 retorna  $X^{\mathbb{R}}$ 

```

---

Os autores demonstraram experimentalmente a equivalência entre a abordagem de *shapelet* embutida de Ye e Keogh (2009) e a transformada *shapelet* em combinação com o algoritmo de classificação C4.5. O trabalho também avaliou a transformada *shapelet* por meio outros classificadores, tais como 1-vizinho mais próximo (utilizando distância Euclidiana e DTW), naive bayes, redes bayesianas, florestas aleatórias e SVM. Os melhores resultados foram verificados com o classificador SVM.

Pelo fato de separar a tarefa de busca por *shapelets* do processo de construção do modelo de classificação, a transformada *shapelet* difere do método original de Ye e Keogh (2009). A principal motivação dos autores para dividir os dois processos consiste em possibilitar a aplicação de diferentes algoritmos de classificação, e também permitir extensões para outras técnicas de aprendizado de máquina.

### 3.5.4 Clustering shapelets

No trabalho de Hills et al. (2013) os autores propuseram uma extensão da transformada *shapelet* para tratar a existência de *shapelets* similares que representam um mesmo padrão. Os autores argumentaram que, para alguns domínios a proposta de Lines et al. (2012) permite a seleção de *shapelets* similares, e que isso pode implicar na redução da interpretabilidade dos modelos de classificação induzidos, e na perda de *shapelets* que poderiam representar outros padrões. Para tratar esse problema, os autores investigaram a técnica de agrupamento hierárquico como uma etapa de pós-processamento. Por meio da avaliação experimental realizada foi constatado que

é possível reduzir a quantidade de *shapelets* e manter a acurácia dos modelos de classificação. No entanto, os autores ponderaram o fato de essa abordagem introduzir o número de centroides como um novo parâmetro a ser analisado em cada domínio.

Baseando-se na generalização da medida de qualidade proposta no algoritmo original da transformada *shapelet*, também foi explorada a aplicação de outras medidas de qualidade, tais como Kruskal-Wallis e mediana de Mood. Por meio da avaliação experimental realizada, os autores demonstraram a equivalência das medidas propostas em relação às medidas existentes (YE; KEOGH, 2009; LINES et al., 2012).

### 3.5.5 Discriminative shapelets

No trabalho de Yuan, Wang e Han (2014) os autores propuseram duas extensões da abordagem de Hills et al. (2013). A primeira para considerar o problema das *shapelets* similares, foi apresentada uma técnica que consiste em analisar o grau de similaridade entre as *shapelets* e seus respectivos limiares de separação. A principal desvantagem em relação a Lines et al. (2012) é a necessidade do uso de uma medida de qualidade que possa ser associada a um limiar de separação, tal como o ganho de informação.

A segunda extensão foi proposta para reduzir a quantidade de *shapelets* selecionadas, similarmente ao que ocorre na fase 1 da abordagem original da transformada *shapelet* (LINES et al., 2012). Os autores argumentam que a técnica de utilizar validação cruzada de cinco partições pode onerar o custo computacional de tempo, quando a análise envolver grandes conjuntos de séries temporais. Desse modo, a abordagem apresentada consiste em realizar a seleção de *shapelets* por meio da adaptação do algoritmo de cobertura sequencial. Considerando a avaliação experimental realizada, os autores demonstraram a efetividade do método proposto para a redução do número de *shapelets* selecionadas. No entanto, a técnica introduziu o número mínimo de cobertura como um novo parâmetro, o qual implica na necessidade de avaliação específica para cada domínio.

## 3.6 GENERALIZAÇÕES E APLICAÇÕES DE SHAPELETS

Alguns estudos na literatura têm explorado a generalização do conceito de *shapelet* para a utilização nas diversas tarefas de mineração de séries temporais. No trabalho de Xing, Pei e Yu (2012), Xing et al. (2011), a identificação de *shapelets* foi utilizada para classificar séries temporais por meio da análise das porções iniciais das séries, tarefa denominada pelos autores de classificação precoce (*early classification*). Em Ghalwash e Obradovic (2012), Ghalwash, Radosavljevic e Obradovic (2013) os autores exploraram alternativas para estender esse conceito para a classificação de séries temporais multivariadas. Posteriormente, em Ghalwash, Radosavljevic e Obra-

dovic (2014), é proposto um modelo de classificação precoce para estimar o grau de incerteza das *shapelets* em séries temporais com ruídos. Em Zakaria, Mueen e Keogh (2012) os autores introduziram o conceito de *shapelet* aplicado à tarefa de agrupamento de séries temporais, nominado pelos autores como *u-shapelets*. No estudo realizado em Grabocka et al. (2014), os autores apresentaram uma nova abordagem para a identificação de *shapelets*. Ao invés de realizar a busca por subsequências em um conjunto de séries temporais, é proposto um modelo de otimização baseado no erro de uma dada função de classificação.

O conceito de *shapelet* tem sido utilizado com sucesso em várias aplicações de classificação na literatura, tais como reconhecimento de gestos (HARTMANN; LINK, 2010); monitoração de sinais fisiológicos (REISS; WEBER; STRICKER, 2011); climatologia (MCGOVERN et al., 2011); identificação de tipos de marcha (SIVAKUMAR; SHAJINA, 2012); demanda de eletricidade (CHANG et al., 2012; GORDON; HENDLER; ROKACH, 2012); diagnóstico de eletrocardiogramas (GHALWASH; RADOSAVLJEVIC; OBRADOVIC, 2013; GHALWASH; RADOSAVLJEVIC; OBRADOVIC, 2014).

### 3.7 CONSIDERAÇÕES FINAIS

Em face aos inúmeros trabalhos desenvolvidos para a classificação de séries temporais, neste capítulo foram apresentados os principais estudos da literatura, que se dedicaram à construção de classificadores inteligíveis. Em especial, foram descritos os métodos baseados na primitiva *shapelet*, a qual tem apresentado resultados promissores como um descritor de características morfológicas interpretáveis (YE; KEOGH, 2009). No entanto, a maior parte dos estudos que buscaram essa caracterização tem se dedicado à melhoria da acurácia por meio de métodos de classificação não simbólicos (LINES et al., 2012; GRABOCKA et al., 2014), e principalmente, ao desenvolvimento de técnicas eficientes para a identificação de *shapelets* (GORDON; HENDLER; ROKACH, 2012; HE et al., 2012; KEOGH; RAKTHANMANON, 2013; WISTUBA; GRABOCKA; SCHMIDT-THIEME, 2015; GRABOCKA; WISTUBA; SCHMIDT-THIEME, 2015). Enquanto que, o aprimoramento dessa abordagem em modelos de classificação simbólicos, em termos de inteligibilidade e de acurácia, tem sido pouco explorado (MUEEN; KEOGH; YOUNG, 2011; HILLS et al., 2013; YUAN; WANG; HAN, 2014).

No próximo capítulo, são apresentadas algumas das contribuições deste trabalho no que se refere ao uso de *shapelets* para a construção de modelos simbólicos de classificação, visando a inteligibilidade e acurácia dos mesmos.



## 4 EXPLORANDO A TRANSFORMADA *SHAPELET*

### 4.1 CONSIDERAÇÕES INICIAIS

Como apresentado no capítulo anterior, a utilização do conceito de *shapelet*, como um descritor de características morfológicas de séries temporais, vem sendo explorado na literatura nos últimos seis anos. Nesse contexto, a maior parte dos estudos tem se dedicado ao desenvolvimento de técnicas e algoritmos para tornar o processo de busca por *shapelets* eficiente em termos de tempo. No entanto, ainda existem poucos trabalhos dedicados ao objetivo original do conceito, que se refere à construção de classificadores simbólicos, tais como árvores de decisão (YE; KEOGH, 2009).

Em Bagnall et al. (2012), os autores sugerem que em problemas que envolvem a análise de características morfológicas, a transformação das séries temporais em um domínio alternativo de representação pode prover melhor acurácia do que a utilização de algoritmos mais complexos de classificação sobre o domínio do tempo. Adicionalmente, em Lines et al. (2012) e em Hills et al. (2013), os autores argumentam que a separação do processo de identificação de *shapelets* da tarefa de indução da árvore de decisão flexibiliza a adaptação de novas abordagens para a identificação de *shapelets*.

Neste capítulo são apresentados os estudos conduzidos neste trabalho, que tiveram o intuito de aprimorar a qualidade da representação de séries temporais por meio da transformada *shapelet* para a construção de árvores de decisão. Inicialmente, na seção 4.2 são apresentadas três propostas de abordagens baseadas na transformada *shapelet* (LINES et al., 2012), com o objetivo de contornar algumas das desvantagens apresentadas pelo algoritmo original. Na seção 4.3, é apresentada a análise experimental realizada para avaliar as abordagens propostas em relação aos algoritmos baseados na abordagem embutida e em relação aos algoritmos considerados estado da arte para a classificação de séries temporais.

### 4.2 EXTENSÕES DA TRANSFORMADA *SHAPELET*

Como descrito no capítulo anterior, *shapelets* podem ser utilizadas para a extração de características morfológicas a partir de séries temporais. Para isso, em Lines et al. (2012) os autores propuseram a transformada *shapelet* (TS) como uma generalização do algoritmo de Ye e Keogh (2009), com o intuito de separar o processo de identificação de *shapelets* da etapa de indução da árvore de decisão. Contudo, para a aplicação do algoritmo TS é necessária a estimativa de alguns parâmetros, tais como o tamanho mínimo e máximo de subsequências a serem avaliadas e o valor de  $k$  para selecionar as melhores subsequências como *shapelets*. A determinação

desses parâmetros tem efeito direto sobre a qualidade dos modelos de classificação produzidos a partir das *shapelets* selecionadas pelo algoritmo TS. Desse modo, para tornar automático o processo de determinação desses parâmetros, nesta seção são apresentadas três abordagens como propostas alternativas ao algoritmo original TS: abordagem exaustiva, abordagem relaxada e abordagem reduzida.

#### 4.2.1 Abordagem exaustiva – TSk

Na abordagem exaustiva (TSk) é proposta a aplicação do algoritmo TS (LINES et al., 2012) por meio de um processo exaustivo de busca com o intuito de explorar os parâmetros *min* e *max* do algoritmo original. Desse modo, a utilização do algoritmo *EstimarMinMax* (algoritmo 8, seção 3.5.2) deixa de ser necessária. O objetivo dessa modificação consiste em permitir ao processo de busca de *shapelets* considerar todos os tamanhos de subsequências possíveis. Nesse cenário, considera-se o menor tamanho de subsequência significativa como  $min = 3$  e o maior tamanho possível para uma *shapelet* como o tamanho da série temporal avaliada,  $max = m$ . O valor do parâmetro  $k$  é mantido fixo para  $\frac{m}{2}$ , conforme Hills et al. (2013). A motivação para a proposta da abordagem exaustiva reside na possibilidade de que *shapelets* de maior qualidade não sejam consideradas quando se limita de modo inferior ou superior o tamanho das subsequências a serem avaliadas.

#### 4.2.2 Abordagem relaxada – TS\*

No algoritmo original da transformada *shapelet* os autores propuseram utilizar somente as  $k$  melhores *shapelets*, o que pode ser determinado por meio de duas estratégias, uma fixa com  $k = \frac{m}{2}$  e outra automática por meio de um processo de validação cruzada usando o conjunto de treinamento, conforme descrito na seção 3.5.1. Ambas as estratégias propostas baseiam-se no valor do tamanho da série temporal  $m$  para determinar o valor de  $k$ . No entanto, pelo melhor do nosso conhecimento, não existem evidências na literatura que demonstrem a relevância de  $m$  para a determinação do parâmetro  $k$ . Ainda, uma estimativa inadequada do parâmetro  $k$  pode deteriorar de modo significativo a representatividade do conjunto de *shapelets* e consequentemente a qualidade do modelo a ser induzido para um dado domínio (HILLS et al., 2013). Isso deve-se ao fato de que durante o processo de busca das  $k$  melhores *shapelets* pode ocorrer um agrupamento de *shapelets* similares na maior parte ou até mesmo na totalidade das  $k$  *shapelets* selecionadas. Para esse contexto, *shapelets* similares referem-se àquelas que foram extraídas da mesma série temporal ou de diferentes séries temporais, mas que representam a mesma informação morfológica. Desse modo, um conjunto de *shapelets* similares que apresenta os maiores valores de ganho de informação pode ser selecionado em detrimento de outras *shapelets* que poderiam



melhor representar outras classes, que contudo, apresentam um ganho de informação inferior.

Como o exemplo apresentado na figura 14, seja um domínio de séries temporais caracterizado por três classes  $\{c_1, c_2, c_3\}$  e seja  $k = 4$  a quantidade de *shapelets* a serem selecionadas em um universo de 8 *shapelets* identificadas. Considerando que as 4 *shapelets* de maior qualidade permitem caracterizar somente informações das séries temporais da classe  $c_1$ , as *shapelets* que descrevem características das classes  $c_2$  e  $c_3$  não serão consideradas, o que consequentemente irá afetar a qualidade do modelo a ser induzido a partir do conjunto de *shapelets* selecionadas para esse domínio.

$$\underbrace{S_H^1(c_1), S_H^2(c_1), S_H^3(c_1), S_H^4(c_1)}_{4 \text{ melhores shapelets}} \quad \underbrace{S_H^5(c_2), S_H^6(c_3), S_H^7(c_3), S_H^8(c_2)}_{\text{shapelets descartadas}}$$

Figura 14 – Ilustração da desvantagem da abordagem exaustiva.

Desse modo, utilizando como base a TSk, a ideia da abordagem relaxada (TS\*) consiste em simplesmente omitir a determinação do parâmetro  $k$ , de modo que todas as subsequências sejam mantidas durante o processo de busca e também na representação final, com exceção dos casamentos triviais (BUHLER; TOMPA, 2002). O intuito dessa modificação consiste em eliminar a necessidade da estimativa do parâmetro  $k$ , e possibilitar a retenção de *shapelets* que possam melhor representar cada uma das classes do domínio avaliado.

#### 4.2.3 Abordagem reduzida – TS+

Como discutido em Lines et al. (2012), Hills et al. (2013) a seleção de poucas *shapelets* na TS pode não prover informação suficiente para a construção de classificadores representativos para um dado domínio. Enquanto que, a utilização de muitas *shapelets* pode causar um sobreajuste no modelo a ser induzido sobre o conjunto de treinamento, e também diluir a influência de *shapelets* mais representativas, dependendo do classificador a ser utilizado.

A abordagem TS\* descrita na seção anterior torna desnecessária a estimativa do parâmetro  $k$ , e portanto, utiliza todas as subsequências possíveis na representação. Para contornar esse problema, é proposto um processamento posterior visando realizar uma filtragem das *shapelets* identificadas, de modo que um subconjunto menor seja obtido de acordo com algum critério de qualidade. Neste trabalho, esse processamento posterior é realizado por meio da aplicação de algoritmos de seleção de atributos, com o intuito de atingir os seguintes objetivos:

1. Possibilitar a estimativa automática do parâmetro  $k$ ;

2. Manter a separação do processo de identificação das  $k$  melhores *shapelets* em relação ao algoritmo de indução do modelo de classificação;

Para atingir esses objetivos, nesta proposta, o conceito de seleção de atributos (SA) é aplicado, pois permite desde a ordenação de atributos de acordo com algum critério de importância até a busca por um subconjunto ótimo de atributos (LEE, 2005). Assim sendo, dado o escopo do problema a ser tratado na abordagem reduzida, o objetivo (1) pode ser atingido por meio da utilização dos algoritmos relacionados a seleção de subconjuntos de atributos. Adicionalmente, os métodos de SA podem ser estruturados em três categorias (KOHAVI; JOHN, 1997):

**embutida:** nessa abordagem a tarefa de seleção do subconjunto de atributos está embutida no algoritmo de aprendizado. Desse modo, o próprio algoritmo de aprendizado utilizado realiza a seleção dos atributos importantes durante o processo de indução do classificador;

**wrapper:** os algoritmos de SA baseados nessa abordagem utilizam o algoritmo de aprendizado como uma técnica “caixa preta” para estimar a qualidade dos atributos. A cada iteração um subconjunto candidato de atributos é selecionado, a partir do conjunto de treinamento, para a indução do classificador. Esse processo é repetido até que um critério de parada seja atingido e, posteriormente, é selecionado o subconjunto que possibilita o melhor desempenho para o classificador avaliado sobre o próprio conjunto de treinamento;

**filtro:** essa abordagem consiste em filtrar atributos irrelevantes e/ou redundantes conforme alguma medida de importância antes da aplicação do algoritmo de aprendizado. Nesse sentido, os métodos da abordagem filtro não necessitam de qualquer interação com o algoritmo de aprendizado para prover o subconjunto de atributos considerados importantes. Desse modo, a medida de importância a ser utilizada deve se basear unicamente nas propriedades intrínsecas dos dados relacionados aos atributos (LIU; MOTODA, 1998).

Dentre as três abordagens mencionadas acima, a abordagem filtro é a única que não necessita de um algoritmo de aprendizado para realizar a tarefa de seleção de atributos. Portanto, observa-se que a utilização de algoritmos da abordagem filtro pode atender ao objetivo (2) descrito nesta seção. Além desse aspecto, o uso da abordagem filtro também justifica-se pela característica de que essas abordagens se baseiam em propriedades intrínsecas dos dados, não sendo necessária a construção de classificadores para avaliar a importância dos atributos.

Neste trabalho são utilizados três algoritmos de seleção de subconjuntos de atributos baseados na abordagem filtro: *Consistency-based Filter* (CS) (LIU; SETIONO,

1996), *Correlation-based Feature Selection* (CFS) (HALL, 2000) e *Fast Correlation-based Filter* (FCBF) (YU; LIU, 2003), os quais são amplamente conhecidos e utilizados na literatura.

**CS:** consiste em um algoritmo probabilístico que realiza a seleção de subconjuntos com poucos atributos e que apresentam alta consistência em relação à classe. Essa abordagem favorece combinações de atributos que permitem particionar os dados em subconjuntos com alguma classe majoritária. Dependendo do tipo de busca a ser utilizado, esse algoritmo pode apresentar uma complexidade de tempo, para o pior caso, de  $O(Mk^2)$ , no qual  $M$  refere-se a quantidade de exemplos (séries temporais) e  $k$  ao número de atributos (*shapelets* selecionadas).

**CFS:** consiste em um algoritmo que baseia-se na avaliação dos subconjuntos de atributos por meio de medidas de relevância e redundância. Para isso, o CFS avalia os subconjuntos de atributos considerando a capacidade preditiva individual de cada um juntamente com o grau de redundância entre eles. Assim, um maior nível de importância é dado aos subconjuntos de atributos que apresentam alta correlação com a classe e baixa redundância entre si. Como medida de correlação, uma das mais conhecidas é o coeficiente de correlação de Pearson, que é obtido dividindo-se a covariância das variáveis pelo produto dos seus desvios padrões. Em relação a complexidade de tempo, o algoritmo CFS também apresenta complexidade de tempo  $O(Mk^2)$ , para o pior caso.

**FCBF:** por meio desse algoritmo a qualidade dos atributos é analisada em duas etapas separadamente. Inicialmente os atributos são avaliados em relação ao grau de correlação com a classe. Desse modo, apenas atributos considerados relevantes para um dado limiar mínimo são mantidos. Na segunda etapa, os atributos selecionados na etapa anterior são avaliados e removidos em função do grau de redundância. Nesse algoritmo a medida *Symmetrical Uncertainty* (PRESS et al., 1992) é utilizada como métrica de correlação em ambas as etapas. A vantagem do algoritmo FCBF sobre as demais abordagens baseadas na seleção de subconjuntos, reside no fato de que a separação das tarefas de análise de relevância e redundância permite minimizar o tempo da busca por subconjuntos, para  $O(k^2)$  no pior caso.

Como resultado da combinação da abordagem TS\* e os algoritmos de seleção de subconjuntos de atributos descritos acima, são propostos três algoritmos a serem avaliados, TS+CS, TS+CFS e TS+FCBF. Para as três variações foi utilizada a busca heurística com o algoritmo guloso *best-first* e direção de busca *forward*.

### 4.3 AVALIAÇÃO EXPERIMENTAL

Nesta seção, é apresentada a avaliação experimental realizada para analisar o desempenho das propostas descritas neste capítulo. As extensões da abordagem transformada foram avaliadas em relação ao algoritmo original da TS, e em comparação com a abordagem embutida. Para situar os resultados das abordagens propostas em um contexto geral na literatura, também foi utilizado um classificador *baseline* baseado no erro majoritário, e o algoritmo 1NN (usando as distâncias ED ou DTW) como referencial do estado da arte.

#### 4.3.1 Algoritmos utilizados

Os algoritmos utilizados nesta avaliação são:

- Baseados na abordagem embutida: *logical shapelets* (LS) (MUEEN; KEOGH; YOUNG, 2011), e *fast shapelets* (FS) (RAKTHANMANON; KEOGH, 2013);
- Baseados na abordagem transformada: transformada *shapelet* (TS) (LINES et al., 2012) e as abordagens apresentadas neste capítulo (TSk, TS\*, TS+CS, TS+CFS e TS+FCBF);
- Classificador *baseline*: consiste em um classificador que sempre prediz a classe majoritária do conjunto de dados;
- Classificador referencial: o algoritmo 1NN, em conjunto com a medida de similaridade ED ou DTW, é considerado a abordagem estado da arte para a classificação de séries temporais (LINES; BAGNALL, 2015; BATISTA et al., 2013; WANG et al., 2013). Ressalta-se que para esses algoritmos são utilizados os dados brutos das séries temporais, sem quaisquer tipo de transformação.

Os algoritmos baseados na abordagem embutida utilizam um processo de indução da árvore de decisão semelhante ao adotado pelo C4.5. No estudo realizado em Lines et al. (2012), os autores demonstraram experimentalmente a equivalência entre o método de indução da árvore de decisão utilizado pelo algoritmo YK e a transformada *shapelet* em combinação com o algoritmo J48 (implementação do algoritmo C4.5), disponível no *framework* Weka (HALL et al., 2009). Desse modo, para avaliar as propostas em mesmo nível, também é adotado o J48 como algoritmo base de aprendizado para as avaliações realizadas nesta seção.

Em relação aos classificadores considerados *baseline*, estes são importantes referências para identificar a eficácia de novas propostas. A acurácia majoritária consiste em uma medida de aprendizado fornecida por um classificador que sempre prediz

a classe majoritária do conjunto de dados. Novas propostas de algoritmos de aprendizado podem utilizar a acurácia da classe majoritária como uma referência mínima de aprendizado a ser alcançada, apesar de não ser necessária a superioridade em relação ao *baseline* (METZ et al., 2015).

Nesta avaliação, adotamos um classificador para ser utilizado como referência da literatura de séries temporais, que não necessariamente produza um modelo simbólico. Desse modo, é proposta a aplicação do algoritmo 1NN utilizando as distâncias ED e DTW, visto que este tem sido apontado como o método estado da arte na literatura, em termos de acurácia. A ideia do classificador referencial é permitir situar os resultados das abordagens avaliadas nesta seção em relação ao que se tem de melhor na literatura atual.

#### 4.3.2 Descrição dos conjuntos de dados

Os conjuntos de dados utilizados nesta avaliação experimental incluem 27 domínios de séries temporais provenientes do *benchmark* da UCR (KEOGH et al., 2006), o qual é amplamente utilizado e difundido na comunidade de séries temporais.

Tabela 2 – Descrição dos conjuntos de dados utilizados.

#	Conjunto de dados	$w$	$M_{treino}$	$M_{teste}$	$m$
1	<i>Adiac</i>	37	390	391	176
2	<i>Beef</i>	5	30	30	470
3	<i>CBF</i>	3	30	900	128
4	<i>ChlorineConcentration</i>	3	467	3840	166
5	<i>Coffee</i>	2	28	28	286
6	<i>DiatomSizeReduction</i>	4	16	306	345
7	<i>ECG200</i>	2	100	100	96
8	<i>ECGFiveDays</i>	2	23	861	136
9	<i>FaceAll</i>	14	560	1690	131
10	<i>FaceFour</i>	4	24	88	350
11	<i>FacesUCR</i>	14	200	2050	131
12	<i>Fish</i>	7	175	175	463
13	<i>GunPoint</i>	2	50	150	150
14	<i>ItalyPowerDemand</i>	2	67	1029	24
15	<i>Lighting2</i>	2	60	61	637
16	<i>Lighting7</i>	7	70	73	319
17	<i>MedicalImages</i>	10	381	760	99
18	<i>MoteStrain</i>	2	20	1252	84
19	<i>OliveOil</i>	4	30	30	570
20	<i>OSULeaf</i>	6	200	242	427
21	<i>SonyAIBORobotSurface</i>	2	20	601	70
22	<i>SwedishLeaf</i>	15	500	625	128
23	<i>Symbols</i>	6	25	995	398
24	<i>SyntheticControl</i>	6	300	300	60
25	<i>Trace</i>	4	100	100	275
26	<i>TwoLeadECG</i>	4	1000	4000	128
27	<i>TwoPatterns</i>	2	23	1139	82

Todos os conjuntos de dados disponíveis na UCR são estruturados em conjunto de treinamento e de teste, que contemplam séries de diversos tamanhos e quantidade

de classes. Ainda, esse *benchmark* contém séries temporais provenientes de domínios artificiais e do mundo real de diversas áreas.

Do total de 45 conjuntos de dados disponibilizados pela UCR, nesta avaliação experimental são utilizados 27 conjuntos, os quais são os mesmos analisados nas avaliações realizadas para os algoritmos LS, FS em Rakthanmanon e Keogh (2013); 1NN ED e 1NN DTW em Keogh et al. (2006); e inclui 17 conjuntos de dados adotados em Hills et al. (2013) na avaliação de TS. Na tabela 2 são descritas as informações pertinentes de cada conjunto de dados utilizado nesta avaliação experimental. Para cada conjunto são detalhados: a quantidade de classes ( $w$ ), a quantidade de séries temporais no conjunto de treinamento ( $M_{treino}$ ) e no conjunto de teste ( $M_{teste}$ ), e o tamanho das séries temporais ( $m$ ).

#### 4.3.3 Configuração dos experimentos

Os experimentos realizados, utilizando as abordagens propostas neste capítulo, foram organizados em quatro etapas, que são ilustradas na figura 15, na qual as linhas contínuas representam o fluxo dos experimentos realizados.

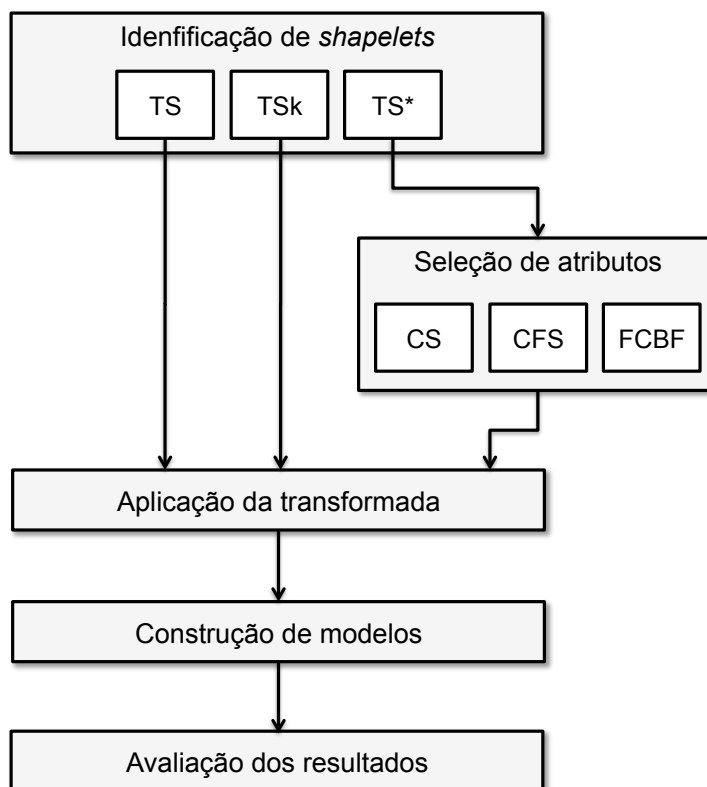


Figura 15 – Representação esquemática da configuração dos experimentos.

**(1) Identificação de *shapelets*:** nessa etapa foi realizado o processo de identificação de *shapelets*, de acordo com cada uma das abordagens propostas.

- (2) **Aplicação da transformada:** com base nas *shapelets* identificadas na etapa (1), nessa etapa os conjuntos de treinamento e de teste foram transformados para a representação atributo-valor.
- (3) **Construção de modelos:** nessa etapa foram induzidos os modelos de classificação por meio do algoritmo J48, o qual foi executado usando os parâmetros com os valores padrão do *framework* WEKA.
- (4) **Avaliação dos resultados:** nessa última etapa, os resultados foram avaliados por meio da estimativa da acurácia para cada um dos modelos construídos usando o método *holdout*, conforme a distribuição dos conjuntos de dados da UCR. Apesar da existência de diversas outras medidas de avaliação, a acurácia foi utilizada como medida de desempenho nesta avaliação, pois tem sido a métrica mais utilizada na comunidade de séries temporais.

Para avaliar as abordagens propostas em relação aos demais algoritmos utilizados nessa avaliação, os experimentos foram conduzidos em duas configurações:

**Avaliação 1:** nesta primeira avaliação foi realizada a análise de desempenho dos algoritmos baseados na transformada *shapelet*. Nesse cenário foram comparadas as modificações propostas neste capítulo (TSk, TS\* e TS+) e o algoritmo original da transformada *shapelet* (TS). Nesta avaliação são utilizados 17 dos 27 conjuntos de dados descritos na tabela 2, os mesmos adotados em [Lines et al. \(2012\)](#) e em [Hills et al. \(2013\)](#).

**Avaliação 2:** na segunda avaliação foram analisadas as abordagens propostas em relação aos algoritmos baseados no conceito de *shapelet* embutida, e em relação ao algoritmo estado da arte 1NN com medida de distância ED e DTW. Para esta análise foram utilizados os 27 conjuntos de dados apresentados na tabela 2.

#### 4.3.4 Análise de significância estatística

Para analisar a existência de diferença estatística entre os desempenhos de  $K$  modelos sobre  $N$  conjuntos de dados, é recomendada a aplicação do teste de Friedman ([DEMSAR, 2006](#)), o qual é uma versão não-paramétrica equivalente ao ANOVA medidas repetidas, para análise de dados pareados. Esse teste baseia-se na comparação de posições de desempenhos (*rank*). Portanto, para cada conjunto de dados, cada um dos modelos avaliados é associado a uma posição, de acordo com seu desempenho, de modo ordenado dos melhores para os piores. Em caso de empates, valores médios de posição são atribuídos. Após, é determinado o *rank* médio para cada um dos  $K$  modelos avaliados. Assim, seja  $R_{ij}$  a posição de desempenho do modelo  $j$ ,



tal que  $1 \leq j \leq K$  para um conjunto de dados  $i$ , tal que  $1 \leq i \leq N$ , o *rank* médio  $R_j$  é calculado conforme a equação 4.1.

$$R_j = \frac{\sum_{i=1}^N R_{ij}}{N} \quad (4.1)$$

Sob a hipótese nula de que o desempenho de todos os modelos comparados são equivalentes, e que portanto seus valores de *rank* médio são iguais, a estatística de Friedman é expressa pela equação 4.2.

$$\chi_F^2 = \frac{12N}{K(K+1)} \left[ \sum_{j=1}^K R_j^2 - \frac{K(K+1)^2}{4} \right] \quad (4.2)$$

No entanto, como discutido em Demsar (2006), recomenda-se a utilização de uma versão menos conservadora do teste de Friedman, proposta em (IMAN; DAVENPORT, 1980), a qual é distribuída de acordo com a distribuição F com  $(K - 1)$  e  $(K - 1)(N - 1)$  graus de liberdade. O cálculo dessa estatística é apresentado na equação 4.3.

$$F_F = \frac{(N - 1)\chi_F^2}{N(K - 1)\chi_F^2} \quad (4.3)$$

Quando a hipótese nula é rejeitada pelo teste de Friedman, isso implica que existe diferença significativa de desempenhos, no entanto não permite discriminar quais modelos apresentam diferença. Nesse cenário, o pós-teste de Nemenyi (NEMENYI, 1963) pode ser utilizado para detectar quais diferenças entre os modelos são significativas (DEMSAR, 2006). De acordo com esse teste, a eficácia de dois métodos é significativamente diferente sempre que seus correspondentes *rank*s médios diferem por pelo menos um determinado valor de diferença crítica (CD), dado pela equação 4.4.

$$CD = q_\alpha \sqrt{\frac{K(K+1)}{6N}} \quad (4.4)$$

Quando a comparação de vários modelos é realizada em relação a um único modelo controle, recomenda-se a aplicação do pós-teste por meio da estatística de Bonferroni-Dunn (DUNN; DUNN, 1961), na qual  $q_\alpha$  pode ser menos restritivo. Nesse caso, somente  $K - 1$  comparações são realizadas, o que confere maior poder estatístico ao pós-teste.

Na tabela 3 são apresentados os valores de  $q_\alpha$  com  $\alpha = 0,05$ , para diferentes quantidades de métodos a serem comparados, utilizando os pós-testes de Nemenyi e de Bonferroni-Dunn.



Tabela 3 – Valores de  $q_{0,05}$  para diferentes valores de  $K$  no pós-teste de Nemenyi e de Bonferroni-Dunn.

K	2	3	4	5	6	7	8	9	10
Nemenyi	1,960	2,343	2,569	2,728	2,850	2,949	3,031	3,102	3,164
Bonferroni-Dunn	1,960	2,241	2,394	2,498	2,576	2,648	2,690	2,724	2,773

#### 4.3.5 Avaliação 1 - Resultados e Discussão

Os resultados dos experimentos realizados, segundo a configuração da Avaliação 1, são apresentados na tabela 4. Cabe ressaltar que os resultados reportados para o algoritmo TS foram extraídos de (HILLS et al., 2013). Os valores são apresentados no formato *acc (rank)*, no qual *acc* corresponde ao valor da acurácia de classificação sobre o conjunto de teste (em termos de porcentagem), e *rank* é a posição relativa do algoritmo em relação aos demais para um dado conjunto de dados. As células destacadas representam os melhores valores de acurácia para um determinado conjunto de dados, considerando todos os algoritmos avaliados. Na última coluna são apresentados os valores de acurácia pela aplicação do classificador *baseline* (Base). Os resultados inferiores ao Base são reportados em negrito. As últimas três linhas representam, para cada algoritmo, os valores de acurácia média, *rank* médio, e quantidade de vezes que apresentou a melhor acurácia em um conjunto de dados (1 vs. todos), respectivamente.

Tabela 4 – Resultados dos experimentos realizados para a Avaliação 1. Os valores são apresentados no formato *acc (rank)*, no qual *acc* corresponde ao valor da acurácia (%) de classificação sobre o conjunto de teste, e *rank* é a posição relativa do algoritmo em relação aos demais para um dado conjunto de dados. As células destacadas representam os melhores valores de acurácia para cada conjunto de dados.

#	TS	TSk	TS*	TS+CFS	TS+CS	TS+FCBF	Base
1	24,30 (6,00)	89,00 (4,00)	87,47 (5,00)	91,56 (3,00)	92,33 (1,00)	91,82 (2,00)	2,05
2	60,00 (6,00)	73,33 (2,00)	73,33 (2,00)	73,33 (2,00)	63,33 (5,00)	70,00 (4,00)	20,00
4	56,48 (6,00)	98,88 (2,00)	98,20 (4,00)	98,33 (3,00)	99,06 (1,00)	98,02 (5,00)	53,26
5	85,71 (4,50)	85,71 (4,50)	85,71 (4,50)	85,71 (4,50)	96,43 (1,50)	96,43 (1,50)	53,57
6	75,16 (1,00)	67,97 (2,00)	66,34 (4,00)	67,65 (3,00)	56,21 (5,00)	47,39 (6,00)	30,07
8	96,17 (6,00)	97,10 (3,00)	97,10 (3,00)	97,10 (3,00)	97,10 (3,00)	97,10 (3,00)	49,71
10	76,14 (1,00)	67,05 (4,00)	42,05 (6,00)	51,14 (5,00)	73,86 (2,50)	73,86 (2,50)	15,91
13	90,67 (6,00)	92,00 (3,00)	92,00 (3,00)	92,00 (3,00)	92,00 (3,00)	92,00 (3,00)	49,33
14	90,96 (5,00)	91,64 (2,50)	91,64 (2,50)	91,64 (2,50)	90,28 (6,00)	91,64 (2,50)	49,85
16	53,42 (3,00)	52,05 (4,50)	47,95 (6,00)	53,42 (2,00)	60,27 (1,00)	52,05 (4,50)	26,03
17	44,87 (6,00)	84,61 (3,00)	83,16 (4,00)	85,53 (1,50)	81,32 (5,00)	85,53 (1,50)	51,45
18	84,42 (6,00)	85,30 (3,00)	85,30 (3,00)	85,30 (3,00)	85,30 (3,00)	85,30 (3,00)	53,91
21	84,53 (3,00)	73,38 (5,00)	73,38 (5,00)	73,38 (5,00)	86,02 (1,50)	86,02 (1,50)	42,93
23	47,14 (6,00)	57,39 (4,50)	57,39 (4,50)	66,33 (3,00)	75,38 (1,00)	74,17 (2,00)	17,39
24	90,33 (6,00)	93,67 (5,00)	95,33 (3,00)	97,33 (1,00)	95,00 (4,00)	97,00 (2,00)	16,67
25	98,00 (2,00)	98,00 (2,00)	98,00 (2,00)	95,00 (6,00)	97,00 (4,50)	97,00 (4,50)	19,00
26	85,25 (6,00)	85,51 (4,00)	85,51 (4,00)	85,51 (4,00)	96,58 (1,50)	96,58 (1,50)	49,96
acc média	73,15	81,92	79,99	81,78	84,56	84,23	
rank médio	4,68	3,41	3,85	3,21	2,91	2,94	
1 vs. todos	3	6	6	7	10	8	

Por meio da aplicação do teste de Friedman com  $\alpha = 0,05$ ,  $K = 6$  e  $N = 17$  sobre os valores de *rank* médio descritos na tabela 4, a hipótese nula de que todos os modelos são equivalentes, em termos de acurácia, foi rejeitada com valores de  $F_F = 2,38$  e  $p\text{-valor} = 0,0458$ . Para verificar quais são as diferenças significativas em relação ao algoritmo TS, o pós-teste de Bonferroni-Dunn foi aplicado com  $\alpha = 0,05$ , o qual apontou valor de diferença crítica de  $CD = 1,65$ .

Como apresentado em Demsar (2006), os resultados dos pós-testes podem ser representados graficamente, como mostrado na figura 16, para os resultados da tabela 4. No gráfico representado nessa figura, a linha principal, no eixo das abscissas, representa o *rank* médio de cada método, na qual valores mais a esquerda representam os melhores desempenhos e os valores mais a direita os piores. As barras abaixo do eixo das abscissas agrupam em *cliques* os modelos que não apresentam diferença estatística com  $\alpha = 0,05$ , em relação ao modelo controle TS. A diferença crítica (CD) é mostrada acima do eixo das abscissas, a partir da esquerda. No caso da representação do pós-teste de Bonferroni-Dunn, somente o *clique* do algoritmo controle é apresentado.

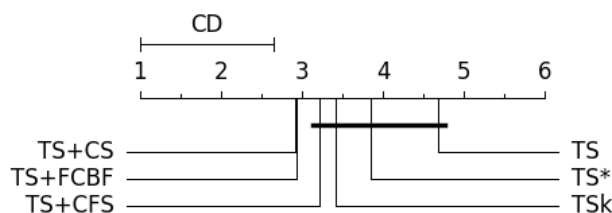


Figura 16 – Diagrama de diferença crítica para o pós-teste de Bonferroni-Dunn.

Com base na análise dos resultados representados na figura 16, pode-se verificar que os modelos TS+CS e TS+FCBF não estão incluídos no *clique* de TS, e portanto apresentam diferenças significativas de desempenho superior em relação ao algoritmo TS. Além desse aspecto, é possível observar que todas as abordagens propostas apresentaram valores de *rank* médio superiores ao TS, em especial as abordagens TS+CS, TS+FCBF e TS+CFS, que apresentaram os maiores valores de *rank* médio. Nota-se também, que somente explorar todos os tamanhos possíveis de subsequências da abordagem exaustiva TSk, ou adicionalmente manter todas as subsequências identificadas, como proposto na abordagem relaxada TS\*, não foram suficientes para demonstrar desempenho superior, estatisticamente, em relação a TS.

### Análise TSk versus TS

Nessa análise é realizada a comparação entre os resultados reportados pela aplicação das abordagens TSk e TS. Na figura 17 são apresentados os resultados comparativos entre ambas, na qual os pontos localizados acima da linha diagonal (região em cor cinza) representam os conjuntos que apresentaram melhor acurácia de TSk em relação a TS. Sobre os 17 conjuntos analisados, a abordagem exaustiva

apresentou melhor acurácia em 11 conjuntos, e empatou com a TS nos conjuntos de dados *Coffee* (#5) e *Trace* (#25).

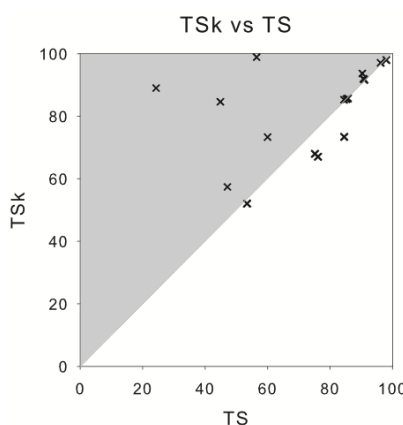


Figura 17 – Comparação entre TSk e TS, em termos de acurácia no conjunto de teste.

Na aplicação do teste Friedman foi constatado que não há diferença estatística entre esses dois métodos. No entanto, pode-se observar que para as conjuntos avaliados, TSk apresentou maior valor de *rank* médio em relação a TS. Em alguns conjuntos houve expressiva melhora na acurácia do modelo de classificação: no conjunto *Adiac* (#1), de 24,30% para 89,00% (diferença de 64,70%); no conjunto *Beef* (#2), de 60,00% para 73,33% (diferença de 13,33%); na conjunto *ChlorineConcentration* (#4), de 56,48% para 98,88% (diferença de 42,40%); no conjunto *MedicalImages* (#17) de 44,87% para 84,61% (diferença de 39,74%). Enquanto que o melhor resultado da TS em relação a abordagem exaustiva foi no conjunto *SonyAIBORobotSurface* (#21) que passou de 84,53% para 73,38% (diferença de 11,15%). A acurácia média de TSk e TS para os 17 conjuntos avaliados foi de 81,92% ( $\sigma = 14,28\%$ ) e 73,15% ( $\sigma = 21,42\%$ ), respectivamente. Sob análise isolada de cada uma dessas abordagens em relação às demais utilizadas na Avaliação 1, TSk apresentou os melhores resultados para 6 conjuntos de dados, enquanto TS foi melhor em 3 conjuntos.

Por meio dos resultados, acima descritos, verifica-se que apesar dos melhores valores de acurácia, obtidos pela aplicação do TSk, não existe diferença significativa, para os dados utilizados, que permita fornecer subsídios para adotar uma das duas estratégias. Contudo, resultados pontuais em alguns dos conjuntos avaliados apresentam um indicativo de que a não exploração de todos os tamanhos de subsequência pode resultar em um impacto negativo no desempenho do método em termos de acurácia.

### Análise TS\* versus TS

Nessa análise é realizada a comparação entre os resultados reportados pela aplicação das abordagens TS\* e TS. Na figura 18 são apresentados os resultados comparativos entre ambas, na qual os pontos localizados acima da linha diagonal

(região em cor cinza) representam os conjuntos que apresentaram melhor acurácia de TS\* em relação a TS. Sobre os 17 conjuntos analisados, a abordagem relaxada apresentou melhor acurácia em 11 conjuntos, empatou com a TS em 2 conjuntos e teve pior acurácia em 4 conjuntos.

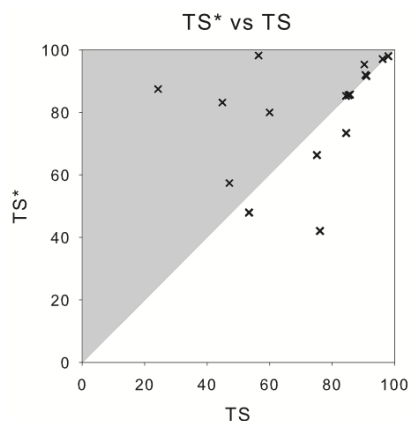


Figura 18 – Comparação entre TS\* e TS, em termos de acurácia no conjunto de teste.

Na avaliação pelo teste Friedman foi constatado que não há diferença estatística entre os resultados reportados para TS\* e TS. No entanto, pode-se observar que para os conjuntos avaliados, TS\* apresentou maior valor de *rank* médio em relação a TS.

As principais diferenças de desempenho superior de TS\* em relação a TS foram registradas para os mesmos conjuntos da análise para o TSk. No conjunto *Adiac* (#1), com 87,47% (diferença de 63,17%); no conjunto *Beef* (#2), com 80,00% (diferença de 20,00%); no conjunto *ChlorineConcentration* (#4), com 98,20% (diferença de 41,72%); no conjunto *MedicalImages* (#17) com 83,16% (diferença de 38,29%). Enquanto que a maior vantagem de TS sobre TS\* foi no conjunto *FaceFour* (#10), para o qual, a acurácia passou de 76,14% para 42,05% (diferença de 34,09%). A acurácia média de TS\* para os 17 conjuntos avaliados foi de 79,99% ( $\sigma = 17,45\%$ ). Sob análise isolada de cada uma dessas abordagens em relação às demais utilizadas na Avaliação 1, TS\* apresentou os melhores resultados para 6 conjuntos de dados.

Por meio dos resultados reportados para essa análise, pode-se verificar que apesar do fato da abordagem relaxada permitir a manutenção de todas as *shapelets* para a representação final da transformada, tal característica não resultou em desempenho superior sobre TS. Além desse aspecto, TS\* foi inferior ao TSk em termos de *rank* médio. Esses resultados reforçam as propriedades discutidas em [Lines et al. \(2012\)](#) e [Hills et al. \(2013\)](#), no que se refere a necessidade de se realizar um processo de seleção das melhores *shapelets*. No entanto, como mencionado na subseção 4.2.3, a determinação das  $k$  melhores *shapelets* (utilizado em TS e TSk) como uma função do tamanho  $m$  da série temporal, não apresenta subsídios na literatura que permitam demonstrar a eficácia da relação.

## Análise TS+ versus TS

Nessa análise é realizada a comparação entre os resultados reportados pela aplicação das abordagens TS+ em relação à abordagem TS. Ressalta-se que as variações TS+CS e TS+FCBF foram as únicas que apresentaram desempenho estatisticamente superior ao método controle TS. Na figura 19 são apresentados os resultados comparativos, na qual os pontos localizados acima da linha diagonal de cada gráfico (região em cor cinza) representam os conjuntos que apresentaram melhor acurácia de TS+ em relação as demais.

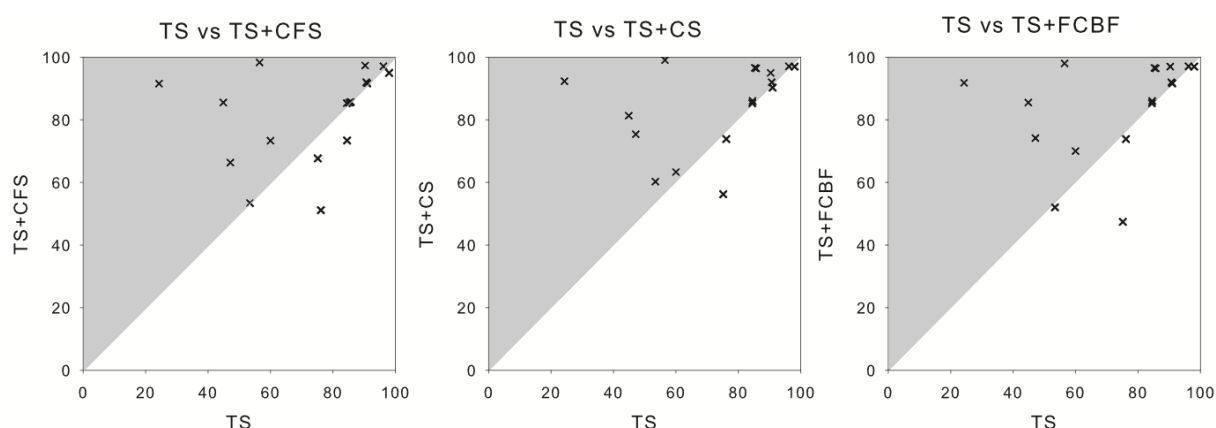


Figura 19 – Comparação entre as variações da abordagem reduzida e TS, em termos de acurácia no conjunto de teste.

Na tabela 5 são apresentados os comparativos em termos de vitórias, empates e derrotas, de cada uma das abordagens TS+CS, TS+CFS e TS+FCBF em relação à abordagem TS.

Tabela 5 – Resultados das variações da abordagem reduzida, em termos de vitórias, derrotas e empates, em relação à abordagem TS.

	TS+CS	TS+CFS	TS+FCBF
Vitórias	13	11	13
Derrotas	4	4	4
Empates	0	2	0

Os valores de acurácia média TS+CS, TS+CFS e TS+FCBF para os 17 conjuntos avaliados foram de 84,56% ( $\sigma = 13,97\%$ ), 81,78% ( $\sigma = 15,00\%$ ) e 84,23% ( $\sigma = 15,74\%$ ), respectivamente. Sob análise isolada de cada uma das abordagens TS+ em relação às demais utilizadas na Avaliação 1, TS+CS apresentou os melhores resultados em 10 conjuntos de dados, TS+CFS em 7 conjuntos e TS+FCBF em 8 conjuntos.

Em alguns conjuntos foram registrados expressivas diferenças em termos da acurácia do modelo de classificação. No comparativo TS+CS e TS, as seguintes melhorias de acurácia foram identificadas: no conjunto *Adiac* (#1), com 92,33% (diferença de

68,03%); no conjunto *ChlorineConcentration* (#4), com 99,06% (diferença de 42,58%); no conjunto *MedicallImages* (#17), com 81,32% (diferença de 36,45%); no conjunto *Symbols* (#23), com 75,38% (diferença de 28,24%). Enquanto que o melhor resultado da TS em relação a TS+CS foi no conjunto *DiatomSizeReduction* (#6) que passou de 75,16% para 56,21% (diferença de 18,95%). No comparativo TS+CFS e TS, as seguintes melhorias de acurácia foram identificadas: no conjunto *Adiac* (#1), com 91,56% (diferença de 67,26%); no conjunto *Beef* (#2), com 73,33% (diferença de 13,33%); no conjunto *ChlorineConcentration* (#4), com 98,33% (diferença de 41,85%); no conjunto *MedicallImages* (#17) com 85,53% (diferença de 40,66%); no conjunto *Symbols* (#23), com 66,33% (diferença de 19,19%). Enquanto que o melhor resultado da TS em relação a TS+CFS foi no conjunto *FaceFour* (#10) que passou de 76,14% para 51,14% (diferença de 25,00%). No comparativo TS+FCBF e TS, as seguintes melhorias de acurácia foram identificadas: no conjunto *Adiac* (#1), com 91,82% (diferença de 67,52%); no conjunto *ChlorineConcentration* (#4), com 98,02% (diferença de 41,54%); no conjunto *MedicallImages* (#17), com 85,53% (diferença de 40,66%); no conjunto *Symbols* (#23), com 74,14% (diferença de 27,03%). Enquanto que o melhor resultado da TS em relação a TS+FCBF foi no conjunto *DiatomSizeReduction* (#6) que passou de 75,16% para 47,39% (diferença de 27,77%).

Como demonstrado, a abordagem reduzida (TS+CS e TS+FCBF) apresentou resultados estatisticamente superiores em relação à abordagem TS, para os 17 conjuntos de dados avaliados. Uma das vantagens da abordagem reduzida consiste na capacidade dos algoritmos de seleção de atributos em analisar características intrínsecas de cada conjunto de dados. Desse modo, as *shapelets* são selecionadas com base nas relações existentes entre elas próprias e com as classes do domínio. Além disso, a abordagem reduzida não necessita da estimativa de parâmetros, como tamanho mínimo e máximo de subsequência, e a quantidade de *shapelets* a serem utilizadas.

#### 4.3.6 Avaliação 2 - Resultados e Discussão

Os resultados experimentais, realizados segundo a configuração da Avaliação 2, são apresentados na tabela 6. Nesse cenário são comparadas as variações apresentadas para a abordagem reduzida (TS+) em relação às abordagens de *shapelet* embutida, e em relação aos algoritmos estado da arte. Ressalta-se que os resultados reportados para os algoritmos LS e FS foram extraídos de [Rakthanmanon e Keogh \(2013\)](#); no qual os experimentos foram conduzidos com parâmetros de tamanho de subsequência  $min = 10$  e  $max = m - 10$ . Os resultados para os algoritmos de 1NN ED e 1NN DTW foram obtidos do trabalho de [Silva, Souza e Batista \(2013\)](#). Os valores são apresentados no formato *acc (rank)*, no qual *acc* corresponde ao valor da acurácia de classificação sobre o conjunto de teste (em termos de porcentagem), e *rank* é a



Tabela 6 – Resultados dos experimentos realizados para a Avaliação 2. Os valores são apresentados no formato *acc (rank)*, no qual *acc* corresponde ao valor da acurácia (%) de classificação sobre o conjunto de teste, e *rank* é a posição relativa do algoritmo em relação aos demais para um dado conjunto de dados. As células destacadas representam os melhores valores de acurácia para cada conjunto de dados.

#	LS	FS	1NN ED	1NN DTW	TS+CFS	TS+CS	TS+FCBF
1	58,60 (6,00)	48,60 (7,00)	61,10 (4,00)	60,40 (5,00)	91,56 (3,00)	92,33 (1,00)	91,82 (2,00)
2	56,70 (4,00)	55,30 (5,00)	53,30 (6,00)	50,00 (7,00)	73,33 (1,00)	63,33 (3,00)	70,00 (2,00)
3	88,60 (5,00)	94,70 (2,00)	85,20 (6,00)	99,70 (1,00)	81,00 (7,00)	90,78 (4,00)	91,89 (3,00)
4	61,80 (6,00)	58,30 (7,00)	65,00 (4,00)	64,80 (5,00)	98,33 (2,00)	99,06 (1,00)	98,02 (3,00)
5	96,40 (3,00)	93,20 (4,00)	75,00 (7,00)	82,10 (6,00)	85,71 (5,00)	96,43 (1,50)	96,43 (1,50)
6	80,10 (4,00)	88,30 (3,00)	93,50 (2,00)	96,70 (1,00)	67,65 (5,00)	56,21 (6,00)	47,39 (7,00)
7	87,00 (2,00)	76,60 (6,00)	88,00 (1,00)	77,00 (5,00)	85,00 (3,00)	83,00 (4,00)	73,00 (7,00)
8	99,40 (2,00)	99,60 (1,00)	79,70 (6,00)	76,80 (7,00)	97,10 (4,00)	97,10 (4,00)	97,10 (4,00)
9	65,90 (6,00)	58,90 (7,00)	71,40 (3,00)	80,80 (1,00)	67,99 (5,00)	72,78 (2,00)	70,06 (4,00)
10	48,90 (7,00)	91,00 (1,00)	78,40 (3,00)	83,00 (2,00)	51,14 (6,00)	73,86 (4,50)	73,86 (4,50)
11	66,20 (7,00)	67,20 (6,00)	76,90 (4,00)	90,49 (1,00)	78,00 (3,00)	69,80 (5,00)	79,85 (2,00)
12	77,70 (7,00)	80,30 (5,00)	78,30 (6,00)	83,30 (4,00)	94,29 (1,00)	90,29 (3,00)	91,43 (2,00)
13	89,30 (7,00)	93,90 (1,00)	91,30 (5,00)	90,70 (6,00)	92,00 (3,00)	92,00 (3,00)	92,00 (3,00)
14	93,60 (3,00)	90,50 (6,00)	95,50 (1,00)	95,00 (2,00)	91,64 (4,50)	90,28 (7,00)	91,64 (4,50)
15	42,60 (7,00)	70,50 (3,00)	75,40 (2,00)	86,90 (1,00)	65,57 (4,50)	50,82 (6,00)	65,57 (4,50)
16	54,80 (5,00)	76,70 (1,00)	57,50 (4,00)	72,60 (2,00)	53,42 (6,00)	60,27 (3,00)	52,05 (7,00)
17	58,70 (6,00)	56,70 (7,00)	68,40 (5,00)	73,70 (4,00)	85,53 (1,50)	81,32 (3,00)	85,53 (1,50)
18	83,20 (6,00)	78,30 (7,00)	87,90 (1,00)	83,50 (5,00)	85,30 (3,00)	85,30 (3,00)	85,30 (3,00)
19	83,33 (6,00)	72,30 (7,00)	86,70 (2,50)	86,70 (2,50)	90,00 (1,00)	86,67 (4,50)	86,67 (4,50)
20	68,60 (3,00)	68,00 (4,00)	51,70 (7,00)	59,10 (6,00)	78,10 (1,00)	64,05 (5,00)	77,27 (2,00)
21	86,00 (3,00)	68,60 (7,00)	69,50 (6,00)	72,50 (5,00)	73,38 (4,00)	86,02 (1,50)	86,02 (1,50)
22	81,30 (4,00)	73,10 (7,00)	78,70 (6,00)	79,00 (5,00)	89,60 (2,00)	90,08 (1,00)	88,80 (3,00)
23	64,30 (7,00)	93,20 (2,00)	90,00 (3,00)	95,00 (1,00)	66,33 (6,00)	75,38 (4,00)	74,17 (5,00)
24	47,00 (7,00)	91,90 (5,00)	88,00 (6,00)	99,30 (1,00)	97,33 (2,00)	95,00 (4,00)	97,00 (3,00)
25	100,00 (1,50)	99,80 (3,00)	76,00 (7,00)	100,00 (1,50)	95,00 (6,00)	97,00 (4,50)	97,00 (4,50)
26	85,60 (6,00)	91,00 (4,50)	91,00 (4,50)	100,00 (1,00)	85,51 (7,00)	96,58 (2,50)	96,58 (2,50)
27	53,90 (7,00)	88,70 (2,00)	74,70 (6,00)	90,40 (1,00)	85,53 (3,00)	84,18 (4,00)	82,23 (5,00)
acc média	73,32	78,71	77,34	82,57	81,68	82,22	82,91
rank médio	5,09	4,46	4,37	3,30	3,69	3,52	3,57
1 vs. todos	1	4	3	10	5	5	3

posição relativa do algoritmo em relação aos demais para um dado conjunto de dados. As células destacadas representam os melhores valores de acurácia para um determinado conjunto de dados, considerando todos os algoritmos avaliados. As últimas três linhas representam, para cada algoritmo, os valores de acurácia média, *rank* médio, e quantidade de vezes que apresentou a melhor acurácia em um conjunto de dados (1 vs. todos), respectivamente.

Por meio da aplicação do teste de Friedman com  $\alpha = 0,05$ ,  $K = 7$  e  $N = 27$  sobre os valores de *rank* médio descritos na tabela 6, a hipótese nula de que todos os modelos são equivalentes, em termos de acurácia, foi rejeitada com valores de  $F_F = 2,59$  e  $p\text{-valor} = 0,0149$ . Desse modo, para verificar quais são as diferenças significativas, o pós-teste de Nemenyi foi aplicado com  $\alpha = 0,05$ , o qual apontou valor de diferença crítica de  $CD = 1,73$ .

Com base na análise dos resultados representados na figura 20, pode-se

verificar que a única diferença significativa, para os 27 conjuntos de dados avaliados, foi entre o algoritmo LS e o 1NN DTW. Contudo, é possível observar que as abordagens TS+CS, TS+FCBF e TS+CFS apresentaram valores de *rank* médio superiores aos algoritmos 1NN ED, FS e LS; enquanto que somente o algoritmo 1NN DTW apresentou valor de *rank* médio acima das abordagens propostas.

Sob análise isolada (1 vs. todos) de cada uma das variações TS+ em relação às demais utilizadas na Avaliação 2, TS+CFS apresentou os melhores resultados para 5 conjuntos de dados, TS+CS também em 5 conjuntos e TS+FCBF em 3 conjuntos. A acurácia média de TS+CFS, TS+CS e TS+FCBF para os 27 conjuntos avaliados foi de 81,68% ( $\sigma = 13,00\%$ ), 82,22% ( $\sigma = 13,86\%$ ) e 82,91% ( $\sigma = 13,59\%$ ), respectivamente, sendo este último o maior resultado de acurácia média dentre os algoritmos avaliados.

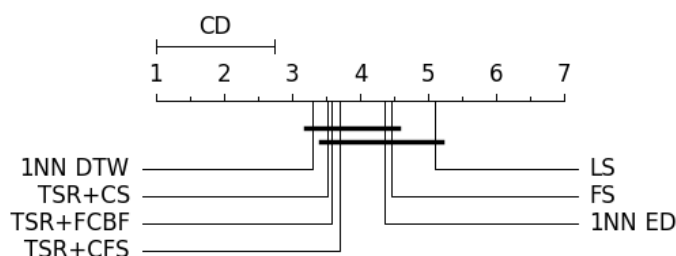


Figura 20 – Diagrama de diferença crítica para o pós-teste de Nemenyi.

Considerando os resultados e análises acima descritos, ressalta-se o fato de que os classificadores construídos utilizando abordagem reduzida são baseados em modelos simbólicos (árvores de decisão), e apresentaram resultados equivalentes aos algoritmos estado da arte 1NN DTW e 1NN ED. Em relação ao algoritmo FS, apesar de também não apresentar diferença significativa em relação às variações da abordagem reduzida TS+, apresentou valor de *rank* médio somente superior ao LS. Além desse aspecto, destaca-se a desvantagem da característica não determinística do algoritmo FS durante o processo de indução da árvore de decisão.

### Análise da abordagem reduzida *versus* abordagem embutida

Nessa análise é realizada a comparação entre os resultados reportados pela aplicação das abordagens TS+ em relação a abordagem embutida (LS e FS). Pela aplicação do teste Friedman não foi verificada diferença estatística na comparação entre as abordagens propostas e os algoritmos LS e FS. No entanto, pode-se observar que, para os 27 conjuntos de dados avaliados, as variações de TS+ apresentaram maiores valores de *rank* médio em relação a LS e FS. Na figura 21 são apresentados os resultados comparativos, na qual os pontos localizados acima da linha diagonal de cada gráfico (região em cor cinza) representam os conjuntos de dados que apresentaram melhor acurácia para TS+.



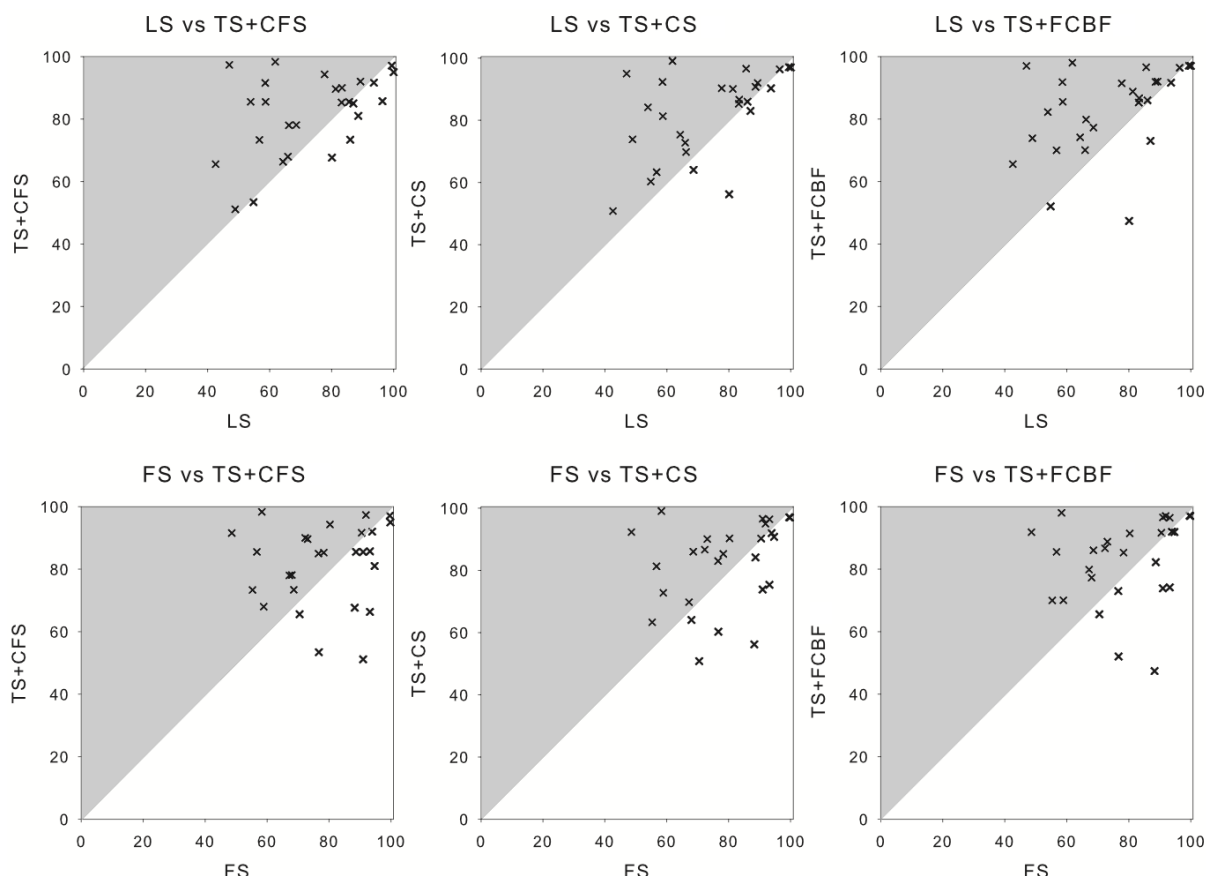


Figura 21 – Comparação entre as variações da abordagem reduzida e os algoritmos da abordagem embutida (LS e FS), em termos de acurácia no conjunto de teste.

O algoritmo TS+CFS apresentou a maior acurácia em 63% e 56% dos conjuntos avaliados, em comparação com LS e FS, respectivamente; TS+CS foi melhor em 78% e 56%; e TS+FCBF foi melhor 78% e 59%. Nenhum empate foi verificado para as comparações realizadas. Na tabela 7 são apresentados os comparativos em termos de vitórias, empates e derrotas, de cada uma das abordagens TS+CFS, TS+CS e TS+FCBF em relação aos algoritmos LS e FS.

Tabela 7 – Resultados das variações da abordagem reduzida, em termos de vitórias, derrotas e empates, em relação a abordagem embutida (LS e FS).

	LS			FS		
	TS+CFS	TS+CS	TS+FCBF	TS+CFS	TS+CS	TS+FCBF
Vitórias	17	21	21	15	15	16
Derrotas	10	6	6	12	12	11
Empates	0	0	0	0	0	0

### Análise da abordagem reduzida *versus* estado da arte

Nessa análise é realizada a comparação entre os resultados reportados pela aplicação das abordagens TS+ em relação aos algoritmos estado da arte. Pela aplicação do teste Friedman não foi verificada diferença estatística na comparação entre

as abordagens propostas e os algoritmos 1NN ED e 1NN DTW. No entanto, pode-se observar que para os 27 conjuntos de dados avaliados as variações de TS+ apresentaram maiores valores de *rank* médio em relação a 1NN ED. Na figura 22 são apresentados os resultados comparativos, na qual os pontos localizados acima da linha diagonal de cada gráfico (região em cor cinza) representam os conjuntos de dados que apresentaram melhor acurácia para TS+.

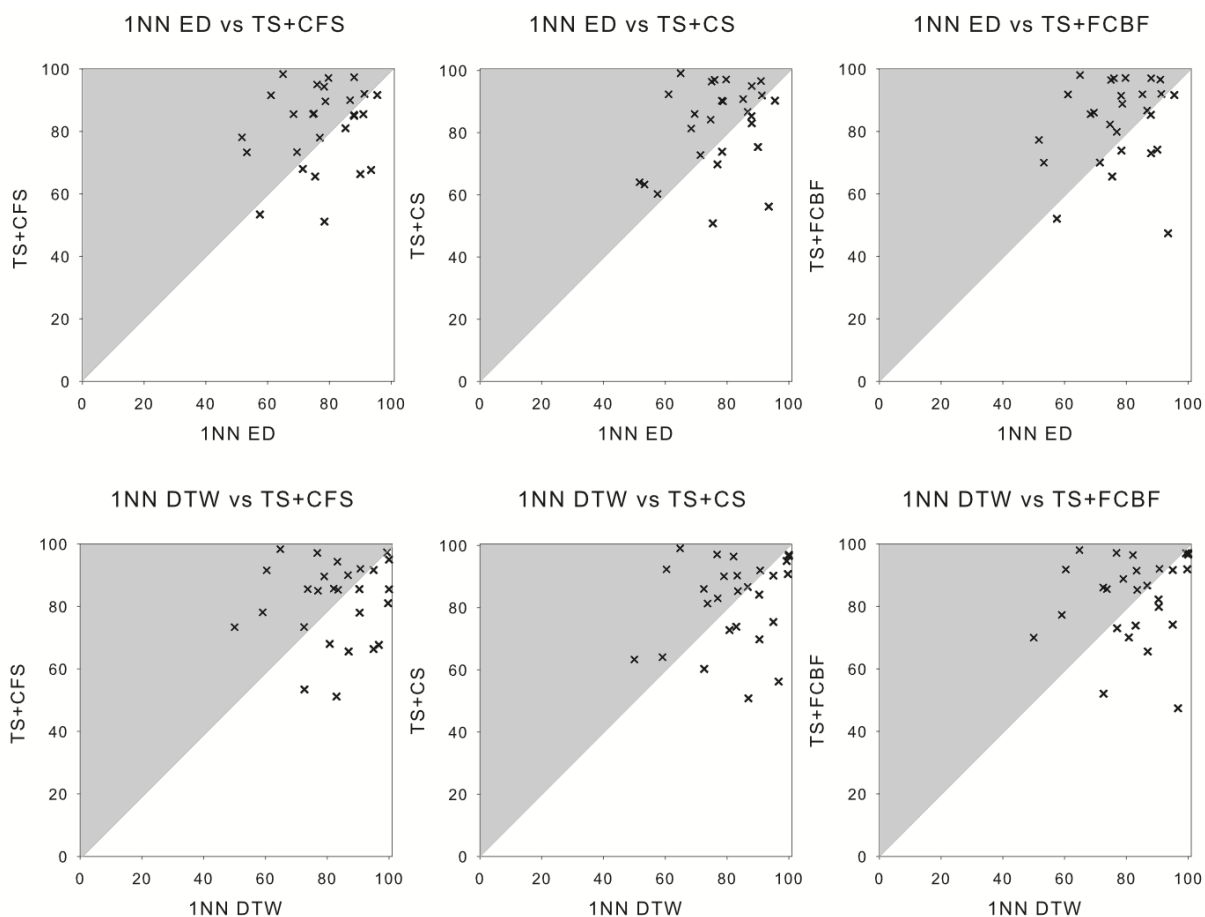


Figura 22 – Comparação entre as variações da abordagem reduzida e os algoritmos estado da arte (1NN ED e 1NN DTW), em termos de acurácia no conjunto de teste.

O algoritmo TS+CFS apresentou a maior acurácia em 59% e 52% dos conjuntos avaliados, em comparação com 1NN ED e 1NN DTW, respectivamente; TS+CS foi melhor em 67% e 48%; e TS+FCBF foi melhor 63% e 44%. Nenhum empate foi verificado para as comparações realizadas. Na tabela 8 são apresentados os comparativos em termos de vitórias, empates e derrotas, de cada uma das abordagens TS+CFS, TS+CS e TS+FCBF em relação a 1NN ED e 1NN DTW.

### **Análise *Texas Sharpshooter Fallacy***

Em Batista, Wang e Keogh (2011) foi apresentado um tipo de análise alternativa dos resultados para problemas de classificação de séries temporais. Os autores propu-

Tabela 8 – Resultados das variações da abordagem reduzida, em termos de vitórias, derrotas e empates, em relação aos algoritmos estado da arte (1NN ED e 1NN DTW).

	1NN ED			1NN DTW		
	TS+CFS	TS+CS	TS+FCBF	TS+CFS	TS+CS	TS+FCBF
Vitórias	16	18	17	14	13	12
Derrotas	11	9	10	13	14	15
Empates	0	0	0	0	0	0

seram avaliar os resultados sob a perspectiva da análise de um gráfico baseado no problema *Texas Sharpshooter Fallacy*. A motivação para esse tipo de análise baseia-se no fato de que muitos estudos na literatura têm reportado resultados comparativos, utilizando a avaliação *holdout* do *benchmark* da UCR; e afirmado que suas respectivas abordagens apresentam desempenho superior, para alguns domínios, somente com base nos resultados obtidos para os conjuntos de teste. Com isso, os autores argumentam que para um determinado algoritmo ser útil é necessário demonstrar *a priori* para quais domínios o mesmo irá apresentar melhor ou pior desempenho em relação a alguma outra abordagem. Para realizar esse tipo de estimativa, em [Batista, Wang e Keogh \(2011\)](#) foi proposto utilizar também a acurácia do conjunto de treinamento para determinar qual método deverá ser aplicado para cada conjunto de dados.

Para a construção do gráfico *Texas Sharpshooter Fallacy*, dois dados são calculados para cada conjunto de dados: o *ganho esperado*, que refere-se a ao *ganho* calculado no conjunto de treinamento (utilizando validação cruzada *leave-one-out*); e o *ganho atual*, que corresponde ao *ganho* calculado sobre o conjunto de teste. Sejam A e B dois métodos a serem comparados nessa análise, o *ganho* de A em relação a B é expresso por:

$$ganho = \frac{acuracia\ método\ A}{acuracia\ método\ B} \quad (4.5)$$

Esse tipo de gráfico pode ser entendido com uma versão da tabela de contingência para valores reais. Desse modo, para um melhor entendimento dos resultados, pode-se dividir o gráfico em quatro regiões, as quais são descritas abaixo:

**VP (verdadeiros positivos):** os pontos registrados dentro dessa região representam os conjuntos de dados para os quais o método A apresentou melhor acurácia que o método B tanto no conjunto de treinamento como no conjunto de teste.

**VN (verdadeiros negativos):** nessa região encontram-se os conjuntos de dados para os quais o método A apresentou acurácia inferior ao método B tanto no conjunto de treinamento como no conjunto de teste.

**FN (falsos negativos):** os conjuntos de dados representados nessa região são aqueles para os quais o método A apresentou acurácia inferior ao método B no conjunto de treinamento, mas apresentou acurácia superior no conjunto de teste.

**FP (falsos positivos):** os pontos registrados dentro dessa região determinam os conjuntos de dados para os quais o método A apresentou melhor acurácia em relação ao método B no conjunto de treinamento, e que no entanto, apresentou acurácia inferior no conjunto de teste.

Na figura 23 são apresentados os gráficos *Texas Sharpshooter Fallacy* construídos para cada uma das comparações realizadas nesta subseção.

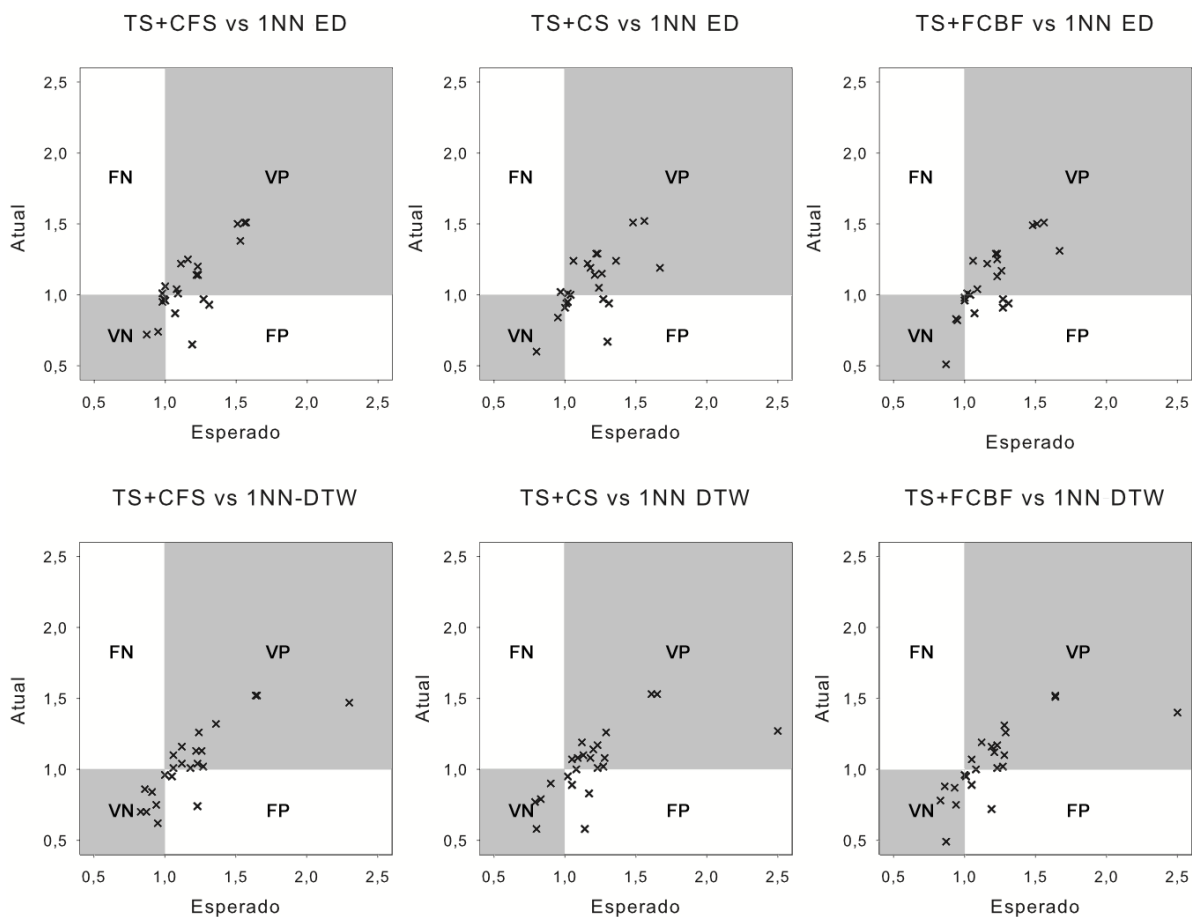


Figura 23 – Comparação entre as variações da abordagem reduzida e os algoritmos estado da arte (1NN ED e 1NN DTW), utilizando a análise *Texas Sharpshooter Fallacy*.

Na tabela 9 estão descritos os resultados registrados para cada uma das quatro regiões. Com base nesses dados, é possível observar que TS+CFS, TS+CS e TS+FCBF tiveram seus desempenhos estimados corretamente para 73%, 68% e 70% dos conjuntos de dados, respectivamente, quando em comparação ao algoritmo 1NN ED. Em relação ao algoritmo 1NN DTW, as estimativas corretas de desempenho foram

87%, 78% e 78%, respectivamente. Deve-se notar que para esta análise foram utilizados 23 dos 27 conjuntos de dados avaliados neste capítulo. Os seguintes conjuntos não foram considerados: *CBF*, *Trace*, *SyntheticControl* e *TwoPatterns*, pois não foram avaliados no trabalho do qual os dados foram coletados.

Tabela 9 – Sumário dos resultados da análise *Texas Sharpshooter Fallacy*.

	1NN ED				1NN DTW			
	VP	VN	FP	FN	VP	VN	FP	FN
TS+CFS	12	4	5	1	14	6	3	0
TS+CS	13	2	6	1	14	4	5	0
TS+FCBF	13	3	7	0	13	5	5	0

Os resultados, acima descritos, evidenciam a capacidade dos algoritmos propostos para a abordagem reduzida, em possibilitar a estimativa correta de seus desempenhos para a maioria dos conjuntos de dados avaliados, quando comparados aos algoritmos estado da arte 1NN ED e 1NN DTW.

#### 4.4 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentadas as adaptações propostas sobre a transformada *shapelet* para contornar as desvantagens do algoritmo original. Dentre as variações propostas, ressalta-se a abordagem reduzida, para a qual foi possível eliminar a necessidade da estimativa dos parâmetros mínimo e máximo de subsequência, bem como do parâmetro  $k$ . Além desses aspectos, por meio da avaliação experimental realizada, foi verificada a superioridade das variações TS+CS e TS+FCBF, em termos de acurácia, sobre o método original da transformada *shapelet*. Em relação aos algoritmos considerados estado da arte da literatura, quando comparados com as variações da abordagem reduzida, não foi verificada diferença estatística. Esses resultados evidenciam a capacidade promissora da transformada *shapelet* para a construção de modelos inteligíveis, por meio de algoritmos de aprendizado simbólico, como árvores de decisão.

Nesse contexto, como demonstrado em [Lines et al. \(2012\)](#) e [Hills et al. \(2013\)](#), a utilização da transformada *shapelet* em combinação com algoritmos de aprendizado não simbólicos favorece a melhoria da acurácia dos modelos, mas deteriora ou impossibilita a inteligibilidade dos classificadores induzidos. Desse modo, no próximo capítulo é introduzido um método para a construção de classificadores híbridos, no qual é possível a aplicação de algoritmos de aprendizado baseado em exemplos, como o 1NN, e a manutenção da representação simbólica.

## 5 ABORDAGEM HÍBRIDA EM DIAGRAMAS DE DECISÃO

### 5.1 CONSIDERAÇÕES INICIAIS

Como descrito no capítulo 3, a utilização de algoritmos baseados no aprendizado de máquina simbólico têm sido o principal meio para a construção de classificadores inteligíveis para séries temporais (BAKSHI et al., 1994; KADOUS, 1999; KUDO; TOYAMA; SHIMBO, 1999; GEURTS, 2001; YAMADA et al., 2003; COTOFREI; STOFFEL, 2002; COTOFREI; STOFFEL, 2005; HIDASI; GASPAP-PAPANEK, 2011; MALETZKE et al., 2014). Adicionalmente, estudos da literatura propuseram a primitiva *shapelet* com o intuito de prover um melhor nível de compreensibilidade para os modelos de classificação induzidos, em especial pela combinação com árvores de decisão (YE; KEOGH, 2009; YE; KEOGH, 2011; MUEEN; KEOGH; YOUNG, 2011; LINES et al., 2012; RAKTHANMANON; KEOGH, 2013; HILLS et al., 2013; YUAN; WANG; HAN, 2014; MALETZKE et al., 2014).

No capítulo 4 foram apresentadas as abordagens propostas neste trabalho para melhorar a qualidade da representação da transformada *shapelet* na construção de modelos baseados em árvores de decisão. No entanto, como mencionado em alguns trabalhos da literatura (DING et al., 2008; BATISTA; WANG; KEOGH, 2011; WANG et al., 2013) e também verificado na avaliação experimental do capítulo 4, o algoritmo 1NN apresenta em geral, o melhor desempenho em termos de acurácia para uma grande variedade de domínios de séries temporais. Porém, esse algoritmo baseia-se em uma estratégia de classificação *lazy*, que ao contrário de um algoritmo *eager*, nenhum modelo é construído explicitamente sobre o conjunto de treinamento. Nesse contexto, neste capítulo é proposto um método (seção 5.3) para a construção de modelos simbólicos de classificação de séries temporais, por meio da descrição de *eventos* que representam características morfológicas. Para implementar esse método, são apresentados os algoritmos *binary shapelet transform* (BST, seção 5.4) e *hybrid instance based decision tree* (HID, seção 5.5).

### 5.2 TRABALHOS RELACIONADOS

Apesar de prover uma representação simbólica por meio de um conjunto de regras estruturadas, a qualidade das predições dos algoritmos tradicionais de indução de árvores de decisão (BREIMAN et al., 1984; QUINLAN, 1986; QUINLAN, 1993) são, frequentemente, afetados por problemas de replicação e de fragmentação (PAGALLO; HAUSSLER, 1990); por dificuldades no tratamento de valores faltantes (FRIEDMAN, 1996) e pela existência de pequenos disjuntos (HOLTE; ACKER; PORTER, 1989). Para

contornar esses problemas, alguns estudos na literatura propuseram a combinação das estratégias *lazy* e modelos simbólicos de classificação. O algoritmo *Lazy Decision Tree* (LazyDT) (FRIEDMAN, 1996; FERN; BRODLEY, 2003) é um exemplo desse tipo de abordagem. Nesse algoritmo uma árvore de decisão é construída para cada novo exemplo a ser classificado. Contudo, apesar de fornecer uma descrição simbólica do conhecimento durante o processo de classificação, nenhum modelo que permita generalizar os dados do conjunto de treinamento é construído explicitamente.

Uma outra estratégia adotada para contornar as desvantagens das árvores de decisão e manter a representação simbólica dos modelos, consiste na construção de hiper-retângulos sobre o conjunto de exemplos. Cada hiper-retângulo pode ser interpretado como uma regra de decisão que determina uma classe, e os limites de cada atributo definem suas bordas. Baseado nesse conceito o algoritmo *Nested Generalised Exemplars* (NGE) (SALZBERG, 1991) utiliza uma abordagem híbrida por meio de hiper-retângulos em combinação com algoritmos de aprendizado baseado em exemplos. Quando um novo exemplo deve ser classificado, primeiramente é verificado se este pode ser coberto por alguma regra existente (hiper-retângulo), caso contrário, o exemplo é associado ao hiper-retângulo mais próximo, segundo alguma métrica de similaridade. Um dos principais algoritmos baseados nessa abordagem é o *Non Nested Generalized Exemplars* (NNGE) (MARTIN, 1995), que foi proposto para tratar problemas relacionados ao aninhamento e sobreposição dos hiper-retângulos do algoritmo NGE (WETTSCHERECK, 1994). Entretanto, apesar dessa estratégia prover uma representação simbólica por meio das regras definidas em hiper-retângulos, o algoritmo NNGE pode ter seu desempenho reduzido devido a possibilidade de sobre-generalizar parte do conjunto de treinamento (GAO; ESTER, 2006). Além desse aspecto, o algoritmo tem como outras desvantagens a utilização de todos os atributos na composição de cada regra; e a possibilidade de apresentar um grande conjunto de regras não ordenadas e parcialmente redundantes para a definição de uma mesma classe (HASPERUE; GIUSTI, 2012).

### 5.3 MÉTODO PROPOSTO

O método proposto neste trabalho consiste em construir modelos de classificação simbólicos para séries temporais, por meio de atributos que permitam representar características morfológicas. Desse modo, a partir deste método é possível prover *a priori* uma representação do modelo na forma de uma árvore de decisão, enquanto que para o processo de classificação pode ser utilizado um algoritmo de aprendizado baseado em exemplos. Sob uma perspectiva geral, o método proposto pode ser estruturado em quatro principais etapas: (1) aprendizado de eventos; (2) representação binária; (3) construção de modelos simbólicos; e (4) classificação. Na figura 24 é apresentada uma



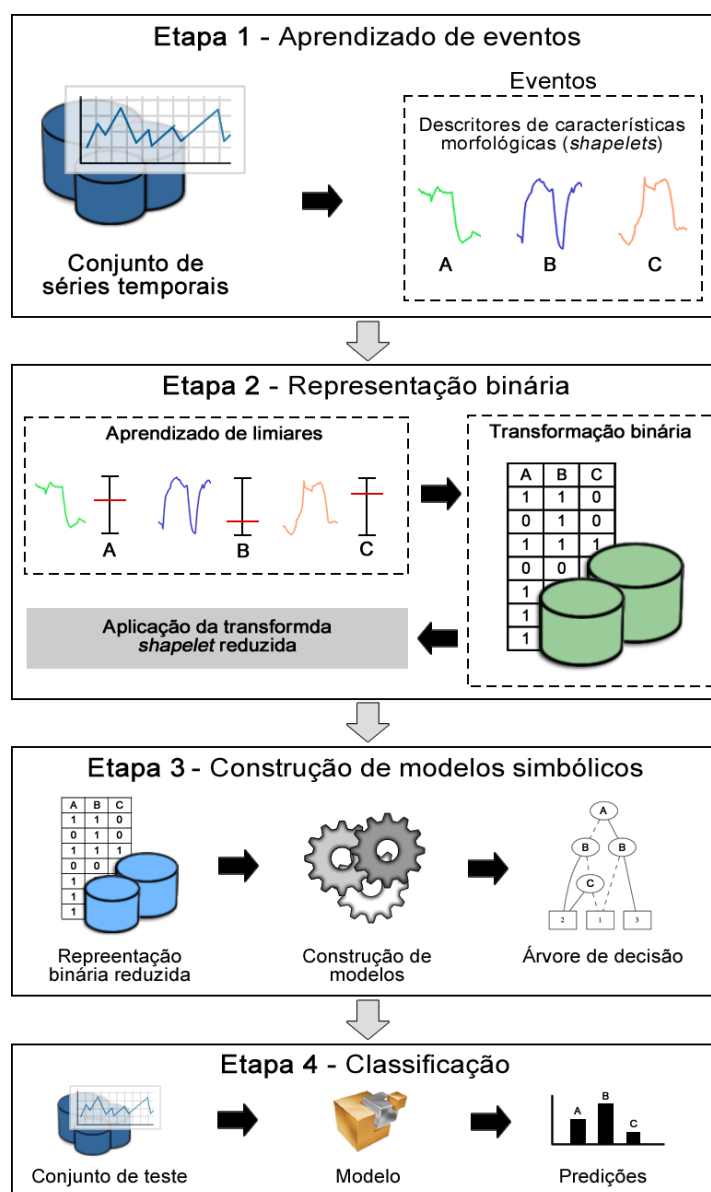


Figura 24 – Representação esquemática do método proposto.

representação esquemática de cada uma dessas etapas.

A primeira etapa é caracterizada pela definição de uma estratégia para a identificação de *eventos* a partir de séries temporais. Considera-se um *evento*, alguma característica que possa ser interpretada em termos de sua *presença* ou *ausência* em uma determinada série temporal. Nesse contexto, como apresentado no capítulo 3 e também explorado no capítulo 4, a primitiva *shapelet* consiste em um descritor de características morfológicas, que tem apresentado efetividade para a construção de modelos simbólicos de classificação de séries temporais. Portanto, neste trabalho, é proposta a utilização da transformada *shapelet* para a caracterização de *eventos* em séries temporais. Mais especificamente, a abordagem relaxada (TS\*), apresentada no capítulo anterior, é adotada nessa primeira etapa.



Para tornar viável a interpretação da transformada *shapelet* em termos de *eventos*, na segunda etapa do método é proposta uma abordagem para converter as *shapelets* em uma representação discreta. Assim, na seção 5.4 é introduzido o algoritmo *binary shapelet transform* (BST), o qual é proposto para prover uma representação binária da transformada *shapelet*.

Na terceira etapa, a representação binária no formato atributo-valor é utilizada para a construção de modelos simbólicos de classificação e, na quarta etapa, os modelos produzidos são utilizados para a classificação de novos exemplos. Essas duas etapas do método são estruturadas no algoritmo proposto *Hybrid Instance based Decision tree* (HID), descrito na seção 5.5.

## 5.4 REPRESENTAÇÃO BINÁRIA

A etapa de representação binária consiste em converter os atributos representados por valores reais da transformada *shapelet*  $X^{\mathbb{R}}$  em atributos unicamente discretos (binários). Desse modo, tem-se que um *evento* expresso por uma *shapelet*, para uma determinada série temporal, pode ser caracterizado por sua *ausência* pelo valor 0 e por sua *presença* pelo valor 1.

Para realizar essa tarefa, nesta seção é apresentado o algoritmo BST, que pode ser estruturado em três principais tarefas: o (1) aprendizado de limiares; a (2) transformação binária; e a (3) representação binária reduzida.

### 5.4.1 Aprendizado de limiares

Reconsiderando os conceitos e definições apresentados na seção 3.4.3, a etapa de aprendizado de limiares consiste em determinar o limiar de separação  $l_s$  (definição 10) para uma dada linha de ordenação  $L$  (definição 9), de modo que o valor de  $l_s$  possibilite a melhor divisão de  $L$  em dois subconjuntos  $L_E$  e  $L_D$ , segundo algum critério de qualidade. Desse modo, a seguir são propostas três estratégias, para determinar o limiar de separação.

#### Precisão – Pr

O processo de determinação de  $l_s$  utilizado por essa estratégia baseia-se em uma abordagem de aprendizado supervisionada, que considera a medida de precisão (equação 5.1) como critério de qualidade. Essa estratégia de estimativa de limiares é descrita no algoritmo 10, o qual é utilizado para identificar o limiar de separação  $l_s$  que possibilita o melhor valor de precisão dado um conjunto de distâncias  $D$ .

**Definição 16 (Precisão)** *Seja uma linha de ordenação  $L$ , um limiar de separação  $l_s$  que divide  $L$  em dois conjuntos disjuntos  $L_1$  e  $L_0$ , um subconjunto de classes  $C_1 \subset C$ , e*

seja  $n_1$  e  $n_{01}$  a quantidade de elementos de  $L_1$  e de  $L$ , respectivamente, que estejam associados a uma classe  $c_i \in C_1$ , a precisão de  $L$  é dada por:

$$Precisao(L, ls) = \sum_{i=1}^w \frac{n_1}{n_{01}} \quad (5.1)$$

---

**Algoritmo 10:** Pr ( $D$ )

---

**Entrada:**  $D$ : conjunto de distâncias associadas a *shapelet*  $S_H^j$

**Saída:**  $ls_{Melhor}$ : limiar de separação

```

1   $maxPREC \leftarrow 0$ ;
2  para  $i \leftarrow 1$  até  $|D| - 1$  faça
3       $ls \leftarrow \frac{1}{2}(d_{i+1} + d_i)$ ;
4       $L \leftarrow D$ ;
5       $PREC \leftarrow Precisao(L, ls)$ ;
6      se ( $PREC > maxPREC$ ) então
7           $maxPREC \leftarrow PREC$ ;
8           $ls_{Melhor} \leftarrow ls$ ;
9      fim
10 fim
11 retorna  $ls_{Melhor}$ 

```

---

### Ganho de informação – Ig

Essa estratégia para a determinação do limiar de separação é a mesma adotada na abordagem de identificação de *shapelets*, tal como descrito no algoritmo 13 (seção 3.4.3). No entanto, nesse caso, o resultado do algoritmo é o valor do limiar de separação  $ls$  que possibilitou o maior ganho de informação.

### Agrupamento – 2k

Para essa estratégia de determinação do limiar de separação, é proposta a aplicação do algoritmo de agrupamento *kmeans*, baseado no aprendizado não-supervisionado. O *kmeans* é um algoritmo tradicional para tarefas que requerem a divisão do espaço de dados visando identificar agrupamentos, no entanto, não garante uma resposta ótima para todas as situações. Desse modo, considerando o fato de que os dados contidos em conjunto de distâncias  $D$  estão representados de modo unidimensional, é proposta a utilização do algoritmo Ckmeans.1d.dp (WANG; SONG, 2011), que oferece a garantia da resposta ótima. Esse algoritmo baseia-se na técnica de programação dinâmica para minimizar a soma dos quadrados das distâncias dentro de cada agrupamento. Nessa estratégia de aprendizado de limiares (algoritmo 11),

são definidos dois agrupamentos para cada  $D$ , os quais representam *presença* (0) e *ausência* (1). Assim, a determinação do limiar de separação é realizada por meio do cálculo da média dos valores dos centróides desses dois agrupamentos.

---

**Algoritmo 11:**  $2k(D)$ 


---

**Entrada:**  $D$ : conjunto de distâncias associadas a *shapelet*  $S_H^j$

**Saída:**  $ls$ : limiar de separação

- 1  $\{centroide^0, centroide^1\} \leftarrow C1kmeans(2, D);$
  - 2  $ls \leftarrow \frac{1}{2}(centroide^0 + centroide^1);$
  - 3 **retorna**  $ls$
- 

#### 5.4.2 Transformação binária

Considerando o processo de aplicação da transformada *shapelet*, descrito na seção 3.5.3,  $X^{\mathbb{R}}$  é a representação atributo-valor do conjunto  $B^M$ , de  $M$  séries temporais para um dado conjunto  $S^k = \{S_H^1, S_H^2, \dots, S_H^k\}$  composto por  $k$  *shapelets*. Para a aplicação do algoritmo BST (algoritmo 12) é necessário definir os seguintes conceitos:

**Definição 17 (Shapelet binária)** *Seja uma shapelet  $S_H^j \in S^k$ ,  $1 \leq j \leq k$ , e seu respectivo conjunto de distâncias  $D_{S_H^j}$  em relação ao conjunto de séries temporais  $B^M$ , uma shapelet  $S_H^j$  é dita binária se  $\forall d_{ij} \in D_{S_H^j}, d_{ij} \in \{0, 1\}, 1 \leq i \leq M$ .*

**Definição 18 (Representação binária)** *Seja  $S^k = \{S_H^1, S_H^2, \dots, S_H^k\}$  um conjunto de  $k$  shapelets binárias, uma representação binária da transformada shapelet  $X^{\mathbb{R}}$  consiste em um conjunto  $X^{\mathbb{B}}$ , tal que  $\mathbb{B} = \{0, 1\}$  e  $\forall T_i \in B^M, S^k \neq \emptyset$ .*

No algoritmo 12, a cada iteração uma *shapelet*  $S_H^j$  é selecionada de  $S^k$  (linha 3), juntamente com seu respectivo conjunto de distâncias  $D_j = \{d_{1j}, d_{2j}, \dots, d_{Mj}\}$ . Após,  $D_j$  é transformado em uma linha de ordenação  $L$ . Na linha 5, o algoritmo *DeterminarLimiar* é utilizado para determinar o valor de  $ls$ , o qual permite dividir a linha de ordenação  $L$  em dois subconjuntos  $L_1$  e  $L_0$ . O subconjunto  $L_1$  representa as séries temporais para as quais a *shapelet*  $S_H^j$  é dita estar *presente*; enquanto que o subconjunto  $L_0$  representa as séries temporais, nas quais  $S_H^j$  está *ausente*. Assim, todo  $d_{ij} \in D_j$  é valorado para 1 ou para 0, dependendo do valor de  $ls_j$  (linhas 7-12). Ao final, a função retorna o conjunto  $X^{\mathbb{B}} = \{D_1 \cup D_2 \cup \dots \cup D_j\}$ , que é a representação binária da transformada *shapelet*, no formato atributo-valor. O algoritmo *DeterminarLimiar* pode utilizar uma das três estratégias de aprendizado de limiares descritas na seção anterior.

---

**Algoritmo 12: BST ( $X^{\mathbb{R}}, S^k$ )**


---

**Entrada:**  $X^{\mathbb{R}}$ : representação atributo-valor da transformada *shapelet*;

$S^k$  : conjunto de  $k$  *shapelets* selecionadas

**Saída:**  $X^{\mathbb{B}}$ : representação binária no formato atributo-valor

```

1  $X^{\mathbb{B}} \leftarrow \emptyset$ ;
2 para cada shapelet  $S_H^i$  em  $S^k$  faça
3    $D_j \leftarrow \{d_{1j}, d_{2j}, \dots, d_{Mj}\}$ ;
4    $L \leftarrow \text{Ordenar}(D_j)$ ;
5    $ls_j \leftarrow \text{DeterminarLimiar}(L)$ ;
6   para cada  $d_{ij}$  em  $L$  faça
7     se  $d_{ij} \leq ls_j$  então
8        $d_{ij} \leftarrow 1$ ;
9     fim
10    senão
11       $d_{ij} \leftarrow 0$ ;
12    fim
13  fim
14   $X^{\mathbb{B}} \leftarrow X^{\mathbb{B}} \cup D_j$ ;
15 fim
16 retorna  $X^{\mathbb{B}}$ 

```

---

#### 5.4.3 Representação binária reduzida

Como discutido no capítulo 4, o desempenho da estratégia original da transformada *shapelet* (TS) depende da estimativa adequada dos parâmetros do algoritmo. Nesse contexto, como também demonstrado pelos resultados da avaliação experimental realizada, uma das abordagens propostas, a transformada *shapelet* reduzida (TS+), apresentou desempenho significativamente superior ao algoritmo original. Portanto, com o intuito de minimizar os problemas apresentados no capítulo anterior, é proposta a aplicação da transformada *shapelet* reduzida após a etapa de transformação binária. Adicionalmente, nessa tarefa é realizada a eliminação de exemplos repetidos que ainda possam estar presentes na representação binária reduzida e que representam classes distintas. Esse processo é aplicado para evitar inconsistências na indução dos modelos.

Apesar da representação binária simplificar a informação agregada pelos valores de distâncias associados a cada *shapelet*, por outro lado algumas vantagens podem ser ressaltadas. A primeira consiste em prover um maior nível de inteligibilidade por meio do uso de atributos unicamente discretos. Uma segunda vantagem consiste na eliminação de possíveis inconsistências conceituais em relação aos modelos construídos utilizando a representação de *shapelets* com valores reais. Sob esse escopo,

uma *shapelet* é dita inconsistente em um modelo de classificação, quando ocorre em mais de uma regra com distintos limiares de separação. Essa característica dificulta a inteligibilidade devido à estratificação dos valores de distâncias associadas a uma determinada *shapelet*.

## 5.5 REPRESENTAÇÃO SIMBÓLICA DOS MODELOS

Nesta seção é apresentado o algoritmo HID, o qual foi proposto neste trabalho para a construção de árvores de decisão híbridas. Esse algoritmo baseia-se em uma representação binária do conjunto de dados, tal como a etapa 2 do método, para a construção de árvores de decisão utilizando diagramas de decisão algébricos.

### 5.5.1 Diagramas de decisão

Seja um conjunto  $\mathbf{x} = \{x_1, x_2, \dots, x_v\}$  composto por  $v$  variáveis booleanas, tal que  $x_i \in \mathbb{B}$ ,  $1 \leq i \leq v$ , uma função booleana é dada como  $f : \mathbf{x} \rightarrow \mathbb{B}$ . Uma das abordagens para a representação de funções booleanas é por meio de um grafo acíclico direcionado (DAG) enraizado, o qual pode ser interpretado como uma árvore de decisão binária (ADB). Nessa estrutura de dados, os vértices terminais são associados a uma constante  $z \in \mathbb{B}$ ; os vértices internos representam alguma variável  $x_i \in \mathbf{x}$  e estão relacionados com duas arestas dirigidas a dois nodos sucessores  $a_1(x_i)$  (quando o valor  $x_i$  é igual a 1) e  $a_0(x_i)$  (quando o valor  $x_i$  é igual a 0). Na representação gráfica da árvore de decisão (figura 25), a aresta  $a_1(x_i)$  é expressa por uma linha contínua e a aresta  $a_0(x_i)$  é expressa por uma linha tracejada.

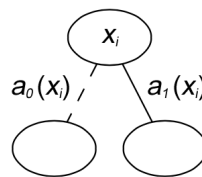


Figura 25 – Representação gráfica da árvore de decisão.

Uma ADB permite a definição do valor de  $f$  sob toda valoração possível das variáveis de  $\mathbf{x}$ . Assim, para cada valoração de  $\mathbf{x}$ , o valor resultante da função  $f$  é dado pelo valor associado ao vértice terminal, o qual pode ser obtido por meio do percurso sobre a ADB, a partir do vértice de entrada (raiz da árvore). Para isso, em cada vértice  $x_i$  é selecionada a aresta correspondente ao valor atribuído a variável  $x_i$  (BARROS, 2007). Como exemplo, seja a função booleana  $f = (x_1 \vee x_2) \wedge (x_2 \vee x_3)$ , na figura 26 (b) é apresentada a representação gráfica da ADB gerada a partir da tabela-verdade 26 (a).

A representação por meio de ADBs está sujeita a existência de caminhos redundantes dentro da estrutura da árvore, devido a replicação de subgrafos isomorfos.

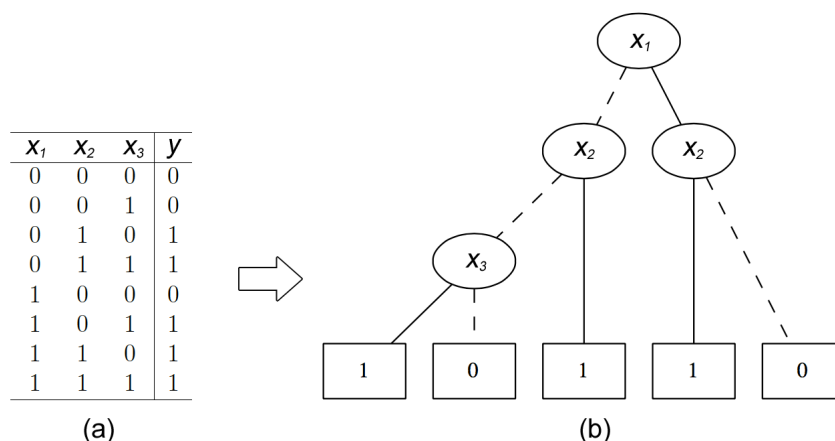


Figura 26 – (a) Tabela-verdade e (b) representação gráfica da ADB para a função  $f = (x_1 \vee x_2) \wedge (x_2 \vee x_3)$

Essa característica implica na construção de uma árvore maior do que seria necessária, e impede a representação da função  $f$  de forma canônica (DELGADO, 2010).

### Diagrama de decisão binário – BDD

Para contornar as desvantagens das ADBs, em Akers (1978) foi proposto o diagrama de decisão binário (BDD), que consiste em uma generalização da ADB pelo fato de permitir que os vértices da árvore possam estar relacionados a mais de um vértice antecessor. Desse modo, torna-se possível o compartilhamento dos subgrafos isomorfos e consequentemente a eliminação de caminhos redundantes dentro da estrutura da árvore (DELGADO, 2010). Posteriormente, em Bryant (1986) foi introduzida a restrição de ordenação das variáveis (diagramas de decisão binários ordenados e reduzidos - ROBDD) para possibilitar a representação de um BDD de forma canônica. Nessa abordagem, os vértices não terminais são estruturados na árvore de decisão por meio de uma ordenação linear fixa. Desse modo, para uma determinada ordenação das variáveis  $\mathbb{O} = \{x_1 \prec x_2 \prec \dots \prec x_v\}$ , qualquer valoração de  $\mathbf{x}$  deve seguir o percurso, sobre a estrutura do BDD, na sequência de verificação dos vértices definida em  $\mathbb{O}$ . Na figura 27 é apresentada a representação gráfica do ROBDD gerado a partir da função booleana  $f = (x_1 \vee x_2) \wedge (x_2 \vee x_3)$ .

A estrutura e o tamanho de um ROBDD para uma dada função  $f$  são características dependentes da ordenação  $\mathbb{O}$  definida para as variáveis de  $\mathbf{x}$ . O processo de busca por uma ordenação ótima para um BDD consiste em um problema de custo computacional exponencial, no pior caso (DELGADO, 2010). No entanto, esse problema tem sido tratado na literatura por meio de algoritmos baseados em heurísticas (DRECHSLER; BECKER; GOCKEL, 1996; BOLLIG; WEGENER, 1996) e na ordenação dinâmica de variáveis (RUDELL, 1993; FUJITA; MATSUNAGA; KAKUDA, 1991), os quais possibilitam manter o tamanho de representação do ROBDD a um custo

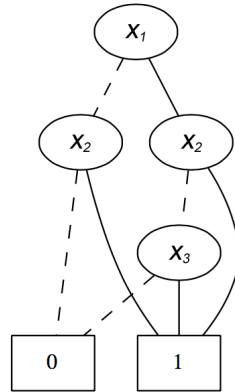


Figura 27 – Representação gráfica do ROBDD da função  $f = (x_1 \vee x_2) \wedge (x_2 \vee x_3)$ .

computacional viável para a maioria das aplicações (BARROS, 2007).

Como um BDD é a representação de uma função booleana  $f$ , as operações válidas para funções são também aplicáveis em BDDs. Se dois BDDs são definidos sobre o mesmo domínio  $\mathbb{B}^v$ , é possível aplicar as operações binárias de conjunção e disjunção. Nesse contexto, uma das vantagens da representação de funções booleanas por meio de um BDD, refere-se ao fato de que o custo computacional das operações sobre essas estruturas não depende do tamanho do espaço de estados, e sim da quantidade de vértices que compõem o diagrama (DELGADO, 2010). Assim, as operações podem ser realizadas diretamente sobre a estrutura dos diagramas, o que permite evitar a enumeração de todas as possíveis valorações de  $\mathbf{x}$ .

#### Diagrama de decisão algébrico – ADD

Um diagrama de decisão algébrico (ADD) (BAHAR et al., 1993) consiste em uma generalização do conceito de BDD, que permite representar uma função  $f : \mathbf{x} \rightarrow \mathbb{R}$ . Desse modo, seja  $y_j$ ,  $1 \leq j \leq 2^v$ , um dos possíveis vértices terminais do ADD que representa a função  $f$ , o valor associado a  $y_j$  é dado por uma constante  $z \in \mathbb{R}$ . Na figura 28 (b) é apresentada a representação gráfica do ADD gerado a partir da tabela descrita na figura 28 (a).

Assim como para BDDs a manipulação de ADDs é realizada por meio de dois algoritmos principais (SOMENZI, 1999; ANDERSEN, 1997):

**Reduzir:** nesse algoritmo é realizado um processo de busca *bottom-up* para a eliminação de redundâncias que possam estar presentes na estrutura de representação do ADD.

**Aplicar:** consiste em um algoritmo de composição de funções sobre ADDs. Sejam dois ADDs  $F$  e  $G$  que representam as funções  $f$  e  $g$ , respectivamente, esse algoritmo

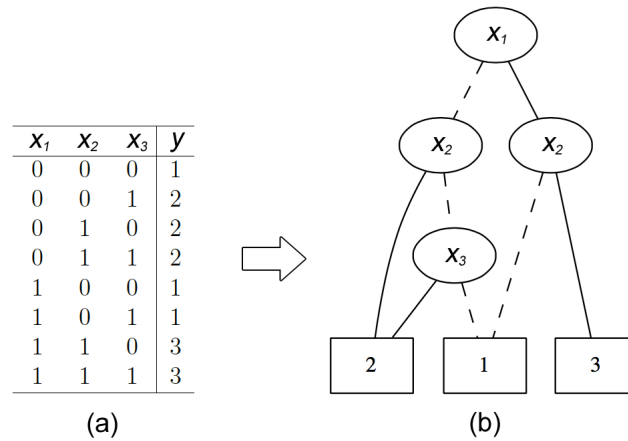


Figura 28 – (a) Tabela e (b) representação gráfica do respectivo ADD.

realiza uma determinada operação  $op$  na forma  $F op G$ , de modo a produzir um ADD resultante consistente com o domínio de  $f$  e  $g$ .

Todas as operações binárias e unárias aplicáveis em um BDD são também possíveis para um ADD. Além disso, pelo fato de possibilitar o mapeamento de funções booleanas para números reais, a aplicação de operações algébricas, tais como soma, diferença, multiplicação, divisão, mínimo e máximo, também é possível sobre ADDs.

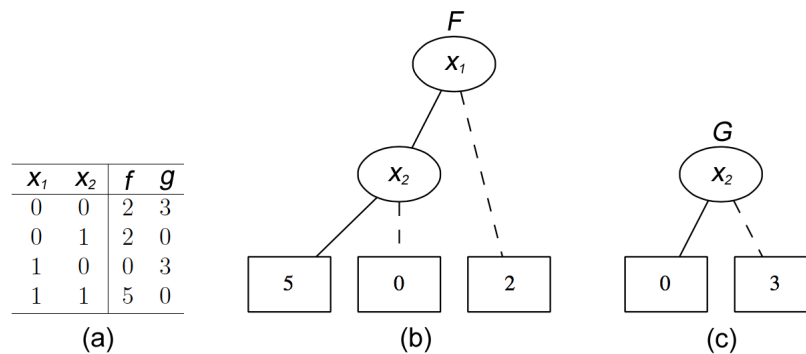


Figura 29 – (a) Tabela das funções  $f$  e  $g$ ; (b) representação gráfica dos ADDs  $F$  e  $G$  (DELGADO, 2010).

Como exemplo, seja a representação gráfica dos ADDs  $F$  e  $G$  apresentada nas figuras 29 (b) e 29 (c), gerados a partir da tabela (figura 29 (a)) das funções  $f$  e  $g$ , respectivamente, os ADDs resultantes das operações de disjunção, conjunção e máximo entre os ADDs  $F$  e  $G$  são apresentados nas figuras 30 (a), 30 (b) e 30 (c), respectivamente.

### 5.5.2 HID: Hybrid Instance based Decision tree

Considerando que, após a etapa 2 do método, cada série temporal passa a ser representada por valores unicamente binários, os conceitos apresentados sobre



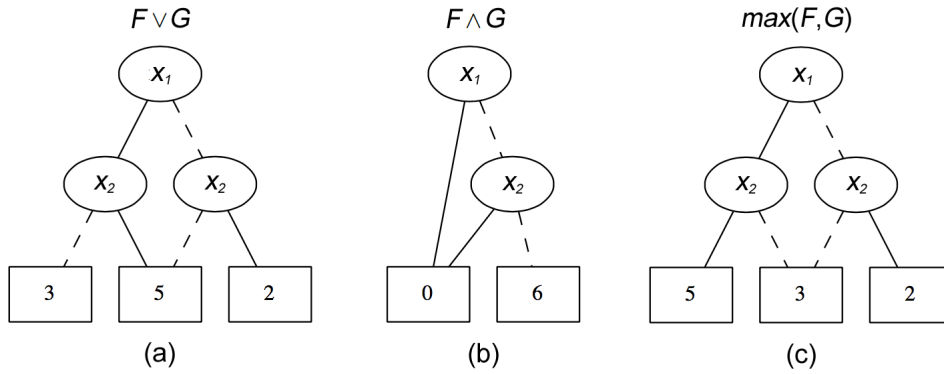


Figura 30 – Representação gráfica das operações de (a) disjunção, (b) conjunção e (c) máximo entre os ADDs  $F$  e  $G$  (DELGADO, 2010).

diagramas de decisão, em especial o ADD, podem ser aplicados para a construção de árvores de decisão a partir da representação produzida pelo algoritmo BST.

Seja uma representação binária  $X^{\mathbb{B}}$  obtida a partir de um conjunto de séries temporais  $B^M$  associado a um conjunto  $C$  de  $w$  classes, é possível interpretar o conjunto de  $k$  *shapelets*  $S^k$  como um conjunto de variáveis booleanas, e cada série temporal  $T_i$  como uma valoração de  $S^k$ . Desse modo, considerando que  $C \subset \mathbb{R}$  pode-se definir  $X^{\mathbb{B}}$  como um ADD por meio de uma função  $A$  tal que:

$$A : S^k \rightarrow C \quad (5.2)$$

No entanto, a função  $A$  que mapeia  $X^{\mathbb{B}}$  para o domínio de classes  $C$  não é conhecida *a priori*. Assim, neste trabalho é proposto o algoritmo HID (algoritmo 13) para a construção do ADD que representa a função  $A$ . Esse algoritmo pode ser estruturado em três principais fases:

- 1. Conjunção de *shapelets*:** nessa fase, cada série temporal  $T_i \in X^{\mathbb{B}}$  é transformada em uma expressão booleana  $E_i$  de acordo com a valoração de  $S^k$ . Caso o valor de uma variável  $S_H^j \in S^k$  seja igual a 0 (falso), essa variável será representada na forma negada  $\neg S_H^j$ . Por exemplo, seja  $T_i = \{0, 1, 0, \dots, 1\}$ , a respectiva expressão booleana é dada por  $E_i = \neg S_H^1 \wedge S_H^2 \wedge \neg S_H^3 \wedge \dots \wedge S_H^k$ .
- 2. Disjunção de exemplos:** nessa fase, todas as expressões de cada classe são combinadas por meio de disjunções. Desse modo, seja  $q$  a quantidade de expressões  $E_i$  para uma determinada classe  $c \in C$ , o conjunto de expressões  $E(c) = \{E_1, E_2, \dots, E_q\}$  é representado pela função  $A(c) = E_1 \vee E_2 \vee \dots \vee E_q$ . Assim, tem-se um ADD que representa  $A(c)$  para cada classe  $c$ , na forma  $A(c) : E(c) \rightarrow \{0, c\}$ .

**3. Composição de classes:** nessa fase, é realizada a construção do ADD que representa a árvore de decisão para  $X^{\mathbb{B}}$ . Esse processo baseia-se na composição sequencial entre os ADDs de cada classe  $c \in C$ . Assim, seja a seguinte ordenação crescente das classes  $c_1 \prec c_2 \prec \dots \prec c_w$ , dada pelos seus respectivos valores (reserva-se o valor 0 para caracterizar o valor booleano *falso*), e sejam os ADDs de cada classe  $A(c)$ :

$$\begin{cases} A(c_1) : E(c_1) \rightarrow \{0, c_1\} \\ A(c_2) : E(c_2) \rightarrow \{0, c_2\} \\ \vdots \\ A(c_w) : E(c_w) \rightarrow \{0, c_w\} \end{cases}$$

a operação máximo (*max*) é aplicada recursivamente do seguinte modo:

$$A = \max_{i=1}^w (A(c_i)) = \max(\max_{i=1}^{w-1} (A(c_i)), A(c_w))$$

Como exemplo, seja um conjunto de  $w = 3$  classes  $C = \{1, 2, 3\}$ , o ADD resultante pela aplicação a operação máximo é dado por:

$$A = \max(\max(A(c_1), A(c_2)), A(c_3))$$

Na figura 31 (c) é apresentado o ADD resultante da operação máximo entre os ADDs  $A(c_1)$  (figura 31 (a)) e  $A(c_2)$  (figura 31 (b)). Na figura 32 (b) é apresentado o ADD resultante da operação máximo entre o ADD  $A(c_3)$  (figura 32 (a)) e o ADD resultante de  $\max(A(c_1), A(c_2))$  (figura 31 (c)).

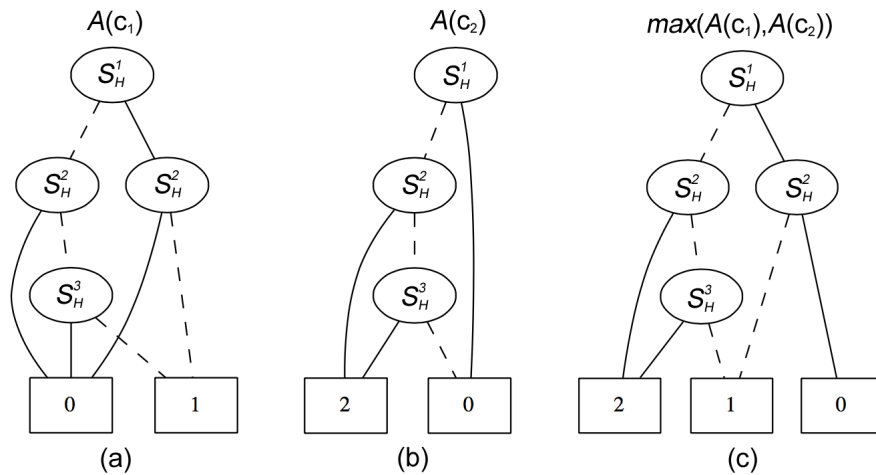


Figura 31 – Representação gráfica do (c) ADD resultante da operação máximo entre os ADDs (a)  $A(c_1)$  e (b)  $A(c_1)$ .

**Algoritmo 13: HID ( $X^{\mathbb{B}}$ )****Entrada:**  $X^{\mathbb{B}}$ : representação binária no formato atributo-valor de  $X^{\mathbb{R}}$ **Saída:**  $A$ : ADD que representa  $X^{\mathbb{B}}$ 

```

1  para cada série temporal  $T_i$  em  $X^{\mathbb{B}}$  faça
2      para cada shapelet  $S_H^j$  em  $S^k$  faça
3           $E_i \leftarrow \emptyset$ ;
4          se  $S_H^j = 1$  então
5               $E_i \leftarrow E_i \wedge S_H^j$ ;
6          fim
7          senão
8               $E_i \leftarrow E_i \wedge \neg S_H^j$ ;
9          fim
10     fim
11      $c \leftarrow \text{Classe}(T_i)$ ;
12      $E(c) \leftarrow E(c) \cup E_i$ ;
13 fim
14 para cada classe  $c_i$  em  $C$  faça
15      $A(c_i) \leftarrow \emptyset$ ;
16     para cada expressão  $E$  em  $E(c_i)$  faça
17          $A(c_i) \leftarrow A(c_i) \vee E$ ;
18     fim
19 fim
20  $A \leftarrow \max_{i=1}^w (A(c_i))$ ;
21 retorna  $A$ 

```

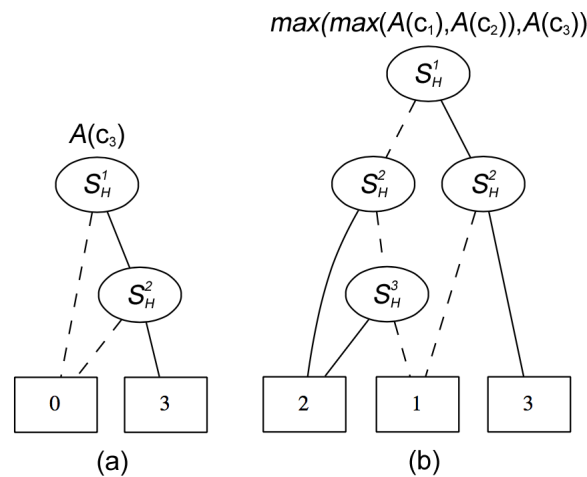


Figura 32 – Representação gráfica do (b) ADD resultante da operação máximo entre o ADD (a)  $A(c_3)$  e o ADD  $\max(A(c_1), A(c_2))$  da figura 31 (c).

Dependendo da quantidade *shapelets* e dos exemplos que caracterizam o conjunto de dados representado por  $X^{\mathbb{B}}$ , dois tipos de modelagem são possíveis para o algoritmo HID, completa ou parcial.

## Modelagem completa

A construção de uma árvore de decisão na modelagem completa envolve a necessidade da existência ou a enumeração de toda a tabela. Em geral, poucos domínios poderão apresentar a representação de todas as valorações possíveis em  $X^{\mathbb{B}}$  diretamente após a aplicação do algoritmo BST. Desse modo, uma alternativa proposta para esse cenário, consiste em utilizar uma função de aproximação  $f'$  que possibilite a estimativa *a priori* das valorações faltantes. No entanto, trata-se de uma abordagem de complexidade exponencial, visto que, seja  $k$  a quantidade de *shapelets*, existem  $2^k$  valorações possíveis. Assim, dependendo do valor de  $k$ , essa abordagem pode se tornar inviável em termos de aplicação computacional. Além desses aspectos, algumas valorações da tabela-verdade podem não representar situações reais no contexto do domínio.

## Modelagem parcial

Considerando o problema da inviabilidade de aplicação da modelagem completa em muitos domínios, neste trabalho é proposto um processo de modelagem parcial. Na construção de um modelo parcial não é necessária a existência ou enumeração de todas as valorações da tabela. Nesta proposta, é realizada a construção do modelo de classificação, para o qual as valorações não descritas na tabela podem ser associadas aos exemplos que estão presentes em  $X^{\mathbb{B}}$ , por meio de uma função de aproximação  $f'$ .

Como exemplo, considerando a tabela-verdade apresentada na figura 28 (a) e a análoga representação de  $X^{\mathbb{B}}$ , na figura 33 (b) é apresentado o modelo parcial construído a partir da tabela parcial, como descrito na figura 33 (a).

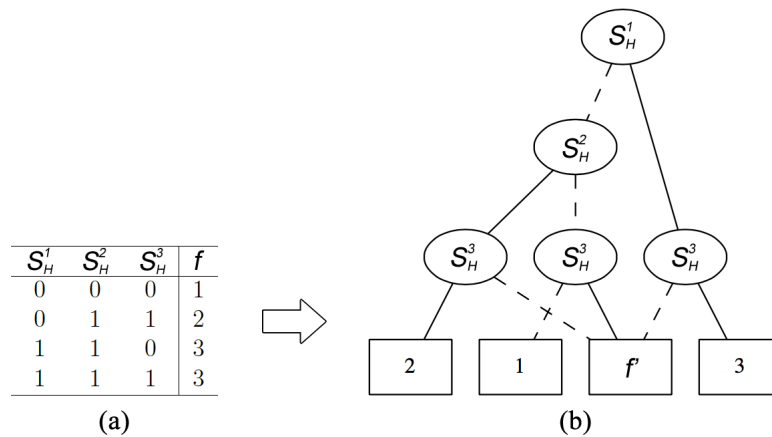


Figura 33 – (a) Tabela parcialmente descrita e (b) representação gráfica do modelo parcial.

### 5.5.3 Classificação

A última etapa do método proposto neste trabalho, consiste em utilizar o modelo simbólico construído, para a classificação de novas séries temporais. Considerando a abordagem de modelos completos, tem-se um algoritmo *eager* de classificação. Desse modo, dada uma ordenação  $\mathbb{O}$  do conjunto de *shapelets* binárias  $S^k$ , basta realizar o percurso sobre a árvore de decisão  $A$  até que um vértice terminal seja encontrado.

Na abordagem de construção de modelos parciais, tem-se um algoritmo híbrido de classificação que pode ser *eager* ou *lazy* dependendo do exemplo a ser classificado. Desse modo, dada uma nova série temporal  $T$ , descrita por  $S^k$ , esta pode ser classificada com base na árvore de decisão  $A$ , caso o percurso sobre a árvore de decisão não encontre o vértice terminal que expressa a função  $f'$ . Caso contrário, significa que o exemplo  $S^k$  não está mapeado em  $X^{\mathbb{B}}$ , e portanto a função de aproximação  $f'$  deve ser utilizada para a classificação. Neste trabalho, é proposta a utilização do algoritmo 1NN como função de aproximação  $f'$  (sobre o conjunto de treinamento), e portanto nesse cenário tem-se a adoção de uma estratégia *lazy* de classificação. Adicionalmente, mesmo para um exemplo  $S^k$  que não esteja mapeado em  $X^{\mathbb{B}}$ , é possível associá-lo a algum caminho existente na árvore, por meio da aproximação provida pelo algoritmo 1NN.

## 5.6 AVALIAÇÃO EXPERIMENTAL

Nesta seção, é apresentada a avaliação experimental realizada para analisar o desempenho do método descrito neste capítulo. Inicialmente, o algoritmo HID foi comparado com um algoritmo tradicional de indução de árvores de decisão (J48) e com um algoritmo híbrido (NNGE), por meio da representação binária apresentada no método proposto. Posteriormente, para situar os resultados em um contexto geral na literatura, o método proposto foi avaliado em relação aos algoritmos baseados na primitiva *shapelet*, tais como a abordagem embutida e a abordagem reduzida (descrita nos capítulos anteriores); e em relação ao algoritmo 1NN (utilizando os dados originais das séries temporais), como referencial do estado da arte.

### 5.6.1 Descrição dos conjuntos de dados

Os conjuntos de dados utilizados nesta avaliação experimental incluem os 27 domínios de séries temporais descritos na seção 4.3.2 (capítulo 4). Adicionalmente, para esta avaliação foram incluídos outros cinco conjuntos, também provenientes do *benchmark* da UCR (KEOGH et al., 2006), totalizando 32 conjuntos.

Nas tabelas 2 (capítulo 4) e 10 são descritas as informações pertinentes a esses conjuntos. Para cada conjunto são detalhados: a quantidade de classes ( $w$ ), a

Tabela 10 – Descrição do conjunto adicional de dados utilizado.

#	Conjunto de dados	$w$	$M_{treino}$	$M_{teste}$	$m$
28	50words	50	450	455	270
29	CricketX	12	390	390	300
30	CricketY	12	390	390	300
31	CricketZ	12	390	390	300
32	WordsSynonyms	25	267	638	270

quantidade de séries temporais no conjunto de treinamento ( $M_{treino}$ ) e no conjunto de teste ( $M_{teste}$ ), e o tamanho das séries temporais ( $m$ ).

### 5.6.2 Algoritmos utilizados

Os algoritmos utilizados nesta avaliação são:

- Indutor base J48: representação binária proposta (BST) utilizando o algoritmo J48 como indutor base;
- Indutor base NNGE: representação binária proposta (BST) utilizando o algoritmo NNGE como indutor base;
- Indutor base HID: representação binária proposta (BST) utilizando o algoritmo HID como indutor base;
- Baseados na abordagem embutida: algoritmo *logical shapelets* (LS) (MUEEN; KEOGH; YOUNG, 2011), e *fast shapelets* (FS) (RAKTHANMANON; KEOGH, 2013);
- Baseados na abordagem transformada: variações da abordagem reduzida (TS+CS, TS+CFS e TS+FCBF), apresentadas no capítulo anterior, utilizando o algoritmo J48 como indutor da árvore de decisão;
- Classificador *baseline*: consiste em um classificador que prediz a classe majoritária do conjunto de dados;
- Classificador referencial: algoritmo 1NN em conjunto com a medida de similaridade ED ou DTW. Para esses algoritmos são utilizados os dados originais das séries temporais, sem qualquer tipo de transformação.

O algoritmo J48 (implementação do algoritmo C4.5) foi utilizado como referência entre os algoritmos baseados na indução de árvores de decisão. O algoritmo NNGE foi proposto para ser utilizado nessa avaliação experimental, pois como o algoritmo HID, também emprega uma estratégia híbrida de classificação. Assim como na avaliação

do capítulo anterior, o algoritmo 1NN (utilizando as distâncias ED e DTW) foi adotado como referência da literatura de séries temporais. A ideia do classificador referencial é permitir situar os resultados das abordagens avaliadas nesta seção em relação ao estado da arte da literatura atual. Também foi utilizado para comparação, a acurácia majoritária como classificador *baseline*.

### 5.6.3 Configuração dos experimentos

Os experimentos realizados para avaliar o método proposto neste capítulo foram organizados em cinco etapas, as quais são ilustradas na figura 34. As linhas contínuas representam o fluxo dos experimentos realizados por meio da representação binária proposta (BST) neste capítulo; enquanto que as linhas tracejadas correspondem ao fluxo experimental da transformada *shapelet*, por meio da abordagem reduzida (TS+), como descrito no capítulo anterior.

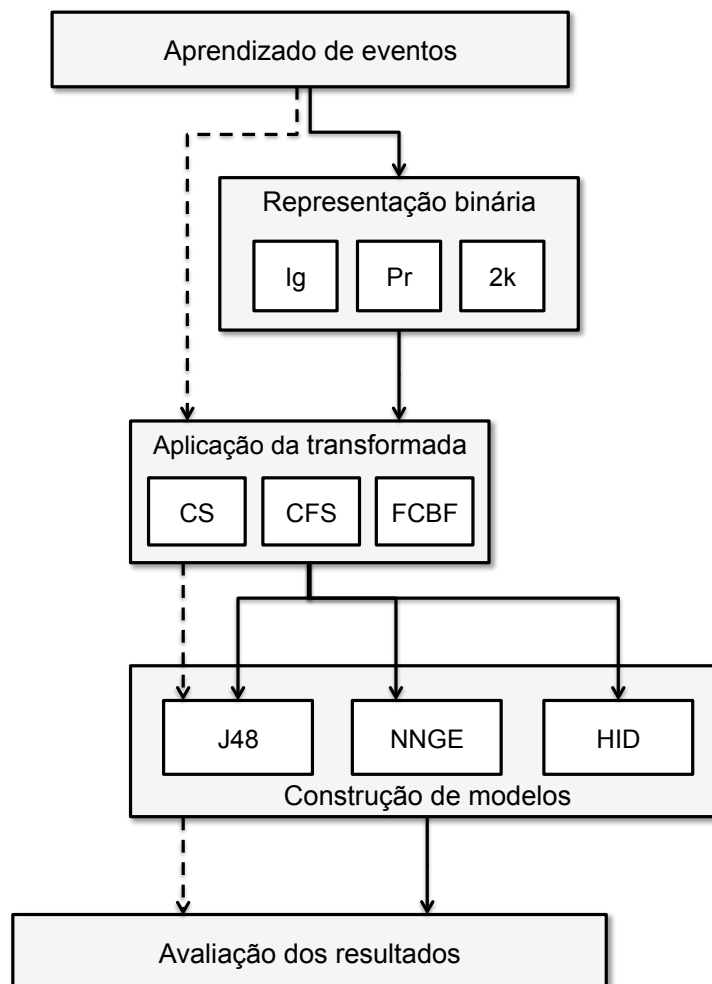


Figura 34 – Representação esquemática da configuração dos experimentos.

**(1) Identificação de eventos:** nessa etapa foi realizado o processo de identificação de eventos por meio da extração de *shapelets*. Para isso, foi aplicada a transformada



*shapelet* segundo os parâmetros definidos para a abordagem relaxada (seção 4.2.2).

- (2) **Representação binária:** nessa etapa, com base nas *shapelets* identificadas na etapa (1), o algoritmo BST foi aplicado utilizando cada um dos três algoritmos de aprendizado de limiares propostos (Pr, Ig e 2k).
- (3) **Aplicação da transformada:** nessa etapa, os conjuntos de treinamento e de teste foram transformados para a representação binária no formato atributo-valor, utilizando cada um dos algoritmos de seleção de atributos empregados na abordagem reduzida (TS+CS, TS+CFS e TS+FCBF) seção 3.5.3.
- (4) **Construção de modelos:** nessa etapa foram induzidos os modelos de classificação para cada um dos algoritmos utilizados nesta avaliação. Exceção dos métodos 1NN ED e 1NN DTW, que são algoritmos *lazy*.
- (5) **Avaliação dos resultados:** nessa última etapa, os resultados foram avaliados por meio da estimativa da acurácia para cada um dos modelos construídos usando o método *holdout*, conforme a distribuição dos conjuntos de dados da UCR. A acurácia foi utilizada como medida de desempenho nessa avaliação, pois tem sido a métrica mais utilizada na comunidade de classificação de séries temporais.

Na etapa de avaliação dos resultados, o método proposto foi comparado em relação aos demais algoritmos utilizados por meio de quatro configurações experimentais, as quais são descritas a seguir:

**Avaliação 1:** nessa primeira avaliação foi explorada a utilização da abordagem *wrapper* de seleção de atributos aplicada aos algoritmos NNGE e HID, para redução da dimensionalidade de representação das *shapelets*. Esses algoritmos foram também avaliados, em termos de acurácia, sem a aplicação de técnicas de seleção de atributos (SSA). Adicionalmente as abordagens *wrapper* e SSA foram analisadas em relação ao algoritmo J48, em termos da quantidade de atributos (*shapelets*) utilizada.

**Avaliação 2:** nessa avaliação foi realizada a análise de desempenho das distintas representações binárias providas pelo algoritmo BST. Para isso os algoritmos de aprendizado de limiares (Ig, Pr e 2k) em combinação com as variações da transformada *shapelet* reduzida (TS+CS, TS+CFS e TS+FCBF) foram analisados por meio dos algoritmos J48, NNGE e HID.

**Avaliação 3:** nessa avaliação foi realizada a análise de desempenho dos modelos simbólicos produzidos pelos algoritmos J48, NNGE e HID, cada qual utilizando a

representação de melhor desempenho verificados na Avaliação 2, em termos de acurácia.

**Avaliação 4:** Nessa avaliação o algoritmo HID foi comparado com as abordagens baseadas na primitiva *shapelet*, já analisadas neste trabalho, tais como a abordagem embutida e a abordagem transformada. Assim, para a abordagem embutida foram selecionados os algoritmos *logical shapelets* (LS) e *fast shapelets* (FS), sendo este último considerado o estado da arte para a construção de modelos simbólicos de classificação de séries temporais. Em relação aos baseados na transformada *shapelet*, foi utilizada a abordagem reduzida TS+, proposta no capítulo 4. Adicionalmente, o algoritmo HID foi também comparado com o algoritmo 1NN, utilizando as medidas de distância ED e DTW, o qual é considerado o estado da arte para a classificação de séries temporais (WANG et al., 2013; DING et al., 2008). Nessa avaliação foram utilizados os 27 conjuntos de dados apresentados na tabela 2 (capítulo 4).

#### 5.6.4 Avaliação 1 – Resultados e discussão

A maioria dos algoritmos baseados na indução de regras ou árvores de decisão utilizam alguma técnica de seleção de atributos durante o processo de indução do modelo, como é o caso do algoritmo J48. Por outro lado, os algoritmos de aprendizado baseados em exemplos, tradicionalmente, não utilizam qualquer tipo técnica para a redução de atributos. Desse modo, considerando os algoritmos híbridos HID e NNGE, que utilizam o algoritmo 1NN como base, nesta avaliação é proposta e analisada a utilização da abordagem de seleção de atributos *wrapper* para esses algoritmos. O objetivo de aplicar essa técnica consiste em uma tentativa de reduzir a quantidade de atributos a ser utilizada para a construção dos modelos, com foco no desempenho dos próprios algoritmos de indução. Um dos principais argumentos apresentados na literatura para a utilização da abordagem *wrapper* é que o mesmo algoritmo que será utilizado para a seleção do subconjunto de atributos também deverá prover uma estimativa de acurácia melhor que algum outro algoritmo, o qual pode possuir um bias de aprendizado diferente (KOHAVI; JOHN, 1997).

Para caracterizar o cenário dessa avaliação, na figura 35 é apresentada a quantidade de atributos<sup>1</sup> (*shapelets*) utilizada pelos modelos construídos por meio dos algoritmos J48, HID e NNGE. Em cada gráfico é descrita a quantidade de atributos para cada representação binária produzida pelos algoritmos de aprendizado de limiares propostos em combinação com as variações da transformada *shapelet* reduzida (TS+). As linhas correspondentes a HID-*wrapper* e NNGE-*wrapper* representam a quantidade de atributos utilizada por meio da aplicação da abordagem *wrapper*, enquanto SSA

<sup>1</sup> Os dados utilizados na representação dos gráficos da figura 35 estão descritos no apêndice A.

refere-se à abordagem na qual a aplicação dos algoritmos HID e NNGE foi realizada sem seleção de atributos.

Por meio da análise dos dados da figura 35 pode-se notar que, para a maioria das representações avaliadas, a quantidade de atributos utilizada pelos algoritmos HID e NNGE, na abordagem SSA, é maior quando em comparação com a abordagem *wrapper* e com o algoritmo J48. As exceções são para as representações binárias geradas pelas estratégias 2k-TS+CS e 2k-TS+FCBF, nas quais as diferenças são menos acentuadas.

Para avaliar o impacto da utilização da abordagem *wrapper*<sup>2</sup> sobre o desempenho dos algoritmos HID e NNGE, cada estratégia de representação binária foi comparada com a abordagem SSA<sup>3</sup>. Para essa análise foi utilizado o teste estatístico *Wilcoxon Signed Ranks* com nível de significância  $\alpha = 5\%$ . Este teste é recomendado para a comparação entre dois modelos quando diversos conjuntos de dados estão envolvidos (DEMSAR, 2006). Nas tabelas 11 e 12 são apresentados os resultados do teste estatístico para cada uma das comparações realizadas para os algoritmos HID e NNGE, respectivamente. Os valores destacados em negrito representam as comparações que apresentaram diferença significativa. Adicionalmente, são apresentados os valores de soma dos *ranks* positivos (+) e negativos (–) que refere-se ao desempenho das abordagens SSA e *wrapper*, respectivamente.

Tabela 11 – Resultados da comparação entre as abordagens SSA e *wrapper* para o algoritmo HID.

	HID-SSA (–) vs HID- <i>wrapper</i> (+)								
	TS+CFS			TS+CS			TS+FCBF		
	lg	2k	Pr	lg	2k	Pr	lg	2k	Pr
<i>p</i> -valor	<b>0,0285</b>	0,0801	<b>0,0177</b>	0,1585	0,3953	0,6030	0,5028	<b>0,0423</b>	0,7278
Soma do <i>rank</i> +	339	317,5	369	30	45	58	160	104,5	189
Soma do <i>rank</i> –	126	147,5	127	75	75	78	116	273,5	162

Tabela 12 – Resultados da comparação entre as abordagens SSA e *wrapper* para o algoritmo NNGE.

	NNGE-SSA (–) vs NNGE- <i>wrapper</i> (+)								
	TS+CFS			TS+CS			TS+FCBF		
	lg	2k	Pr	lg	2k	Pr	lg	2k	Pr
<i>p</i> -valor	<b>0,0012</b>	<b>0,0002</b>	0,1527	0,3472	0,8200	<b>0,0390</b>	<b>0,0155</b>	0,0574	0,1970
Soma do <i>rank</i> +	283	410,5	302	51	20	33	185	233	195
Soma do <i>rank</i> –	42	54,5	163	27	25	3	46	92	105

Como reportado na tabela 11, em três comparações foram verificadas diferenças significativas para o algoritmo HID. Nas estratégias de representação binária

<sup>2</sup> Os valores de acurácia para os algoritmos HID e NNGE, usando a abordagem *wrapper*, são apresentados nas tabelas 15 e 14, respectivamente

<sup>3</sup> Os valores de acurácia para os algoritmos HID e NNGE, usando a abordagem SSA, estão descritos no apêndice B.

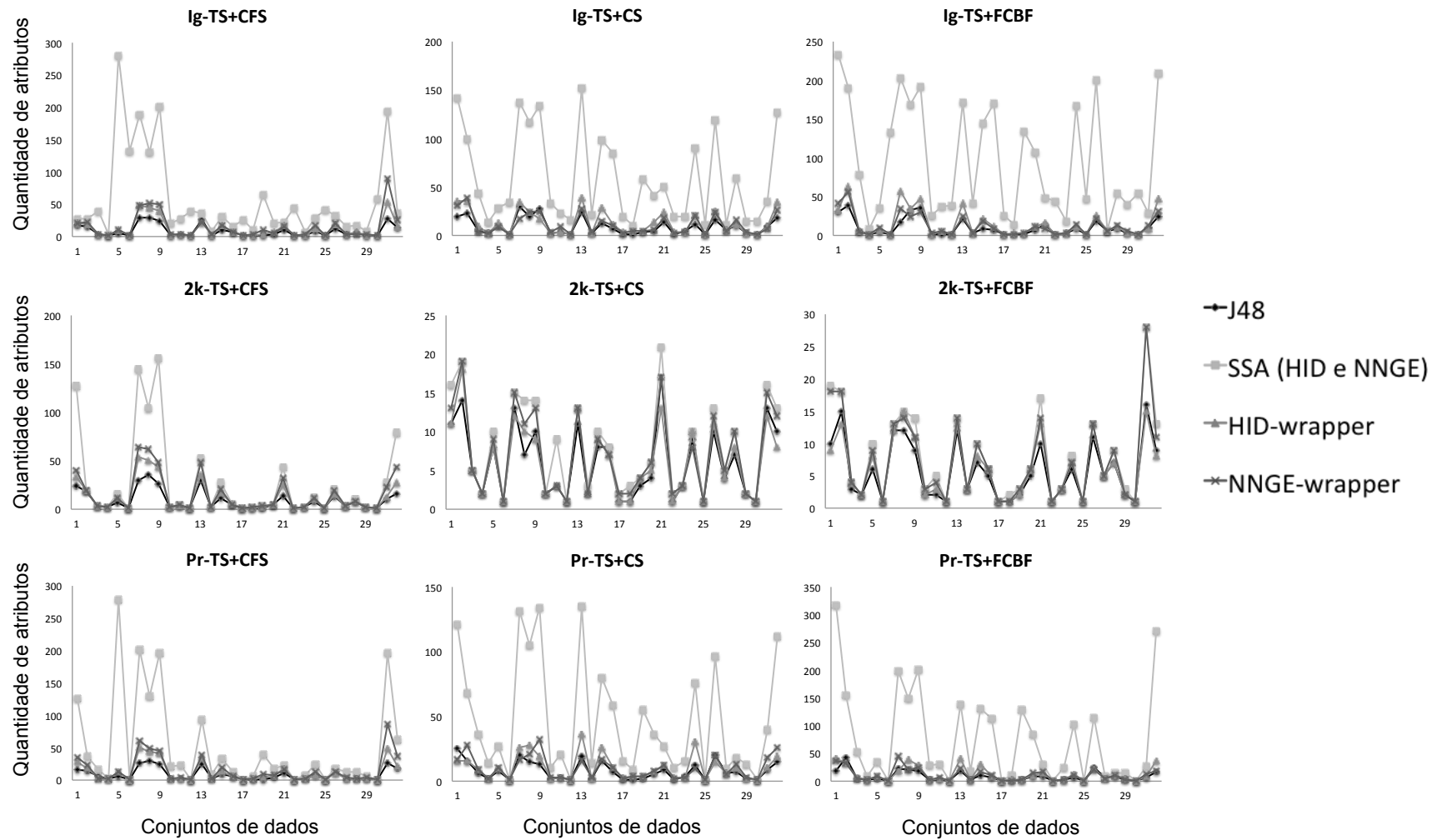


Figura 35 – Quantidade de atributos utilizada pelos algoritmos J48 e, HID e NNGE com abordagem *wrapper* e abordagem SSA, considerando cada uma das estratégias de binarização em cada conjunto de dados avaliado.

Ig-TS+CFS e Pr-TS+CFS o desempenho da abordagem *wrapper* foi inferior a SSA, enquanto na estratégia 2k-TS+FCBF o desempenho foi superior. Em relação aos resultados do algoritmo NNGE descritos na tabela 12, em quatro estratégias de representação binária foram verificadas diferenças significativas, para as quais a abordagem SSA apresentou melhor desempenho.

Por meio da análise dos resultados apresentados nesta avaliação, é possível verificar que na maioria das comparações, a redução de atributos por meio da abordagem *wrapper* não apresentou diferenças significativas de desempenho, enquanto uma menor quantidade de atributos foi utilizada para a construção dos modelos.

#### 5.6.5 Avaliação 2 – Resultados e discussão

Os resultados dos experimentos realizados, segundo a configuração da Avaliação 2, considerando os algoritmos base J48, NNGE e HID, são apresentados nas tabelas 13, 14 e 15, respectivamente. Os algoritmos NNGE e HID foram avaliados por meio da utilização da abordagem *wrapper*, considerando os resultados analisados na Avaliação 1. Os valores são apresentados no formato *acc (rank)*, no qual *acc* corresponde ao valor da acurácia de classificação sobre o conjunto de teste (em termos de porcentagem), e *rank* é a posição relativa do algoritmo em relação aos demais para um dado conjunto de dados. As colunas 2-4, 5-7 e 8-10 correspondem aos resultados reportados para os algoritmos de binarização Ig, 2k e Pr em combinação com a transformada *shapelet* baseada nas abordagens TS+CFS, TS+CS e TS+FCBF, respectivamente. Na última coluna são apresentados os valores de acurácia pela aplicação do classificador *baseline*. As últimas três linhas representam, para cada algoritmo, os valores de acurácia média, *rank* médio, e quantidade de vezes que apresentou a melhor acurácia em um conjunto de dados (1 vs todos), respectivamente. As células destacadas representam os melhores valores de acurácia para um determinado conjunto de dados, considerando todos os algoritmos avaliados.

A análise estatística dos resultados de cada algoritmo base foi realizada por meio do teste de Friedman com nível de significância  $\alpha = 5\%$ ,  $K = 9$  (quantidade de algoritmos) e  $N = 32$  (número de conjunto de dados). Quando constatadas diferenças significativas, o pós-teste de Nemenyi foi aplicado com  $\alpha = 5\%$ . Nessa análise, foi verificado valor de diferença crítica de  $CD = 2, 12$ .

Tabela 13 – Resultados dos experimentos realizados utilizando o J48 como algoritmo base. Os valores são apresentados no formato *acc (rank)*, no qual *acc* corresponde ao valor da acurácia (%) de classificação sobre o conjunto de teste, e *rank* é a posição relativa do algoritmo em relação aos demais para um dado conjunto de dados. As células destacadas representam os melhores valores de acurácia para cada conjunto de dados.

#	TS+CFS			TS+CS			TS+FCBF			baseline
	lg	2k	Pr	lg	2k	Pr	lg	2k	Pr	
1	79,54 (3,00)	85,42 (2,00)	76,98 (4,00)	89,51 (1,00)	73,91 (5,00)	50,64 (9,00)	61,64 (8,00)	65,98 (7,00)	68,03 (6,00)	2,05
2	66,67 (3,50)	66,67 (3,50)	60,00 (7,50)	66,67 (3,50)	60,00 (7,50)	53,33 (9,00)	66,67 (3,50)	73,33 (1,00)	63,33 (6,00)	20,00
3	85,11 (7,00)	91,78 (2,50)	91,11 (4,50)	85,11 (7,00)	91,78 (2,50)	91,11 (4,50)	85,11 (7,00)	91,89 (1,00)	82,22 (9,00)	33,11
4	90,78 (8,00)	95,08 (2,00)	95,00 (3,00)	94,38 (5,00)	95,76 (1,00)	91,48 (7,00)	89,74 (9,00)	94,43 (4,00)	91,98 (6,00)	53,26
5	96,43 (5,50)	100,00 (1,50)	96,43 (5,50)	96,43 (5,50)	100,00 (1,50)	96,43 (5,50)	96,43 (5,50)	92,86 (9,00)	96,43 (5,50)	53,57
6	78,43 (4,00)	76,47 (6,00)	88,24 (1,50)	78,43 (4,00)	73,20 (7,00)	88,24 (1,50)	62,75 (8,00)	54,58 (9,00)	78,43 (4,00)	30,07
7	85,00 (5,00)	77,00 (9,00)	86,00 (3,00)	83,00 (6,00)	82,00 (7,00)	86,00 (3,00)	89,00 (1,00)	79,00 (8,00)	86,00 (3,00)	64,00
8	98,49 (3,50)	97,33 (7,50)	98,49 (3,50)	98,49 (3,50)	97,33 (7,50)	98,49 (3,50)	98,49 (3,50)	96,86 (9,00)	98,49 (3,50)	49,71
9	69,11 (2,00)	66,45 (5,00)	66,63 (4,00)	70,00 (1,00)	67,34 (3,00)	66,39 (6,00)	58,99 (9,00)	63,96 (7,00)	63,61 (8,00)	4,26
10	56,82 (8,50)	84,09 (3,00)	90,91 (1,50)	56,82 (8,50)	82,95 (4,00)	90,91 (1,50)	68,18 (6,50)	81,82 (5,00)	68,18 (6,50)	15,91
11	72,49 (4,00)	75,85 (1,00)	70,29 (5,00)	74,78 (2,00)	66,98 (6,00)	62,73 (8,00)	58,88 (9,00)	74,10 (3,00)	64,44 (7,00)	14,34
12	90,29 (5,00)	90,29 (5,00)	90,86 (2,50)	90,29 (5,00)	90,86 (2,50)	89,14 (7,00)	86,86 (8,00)	86,29 (9,00)	93,14 (1,00)	12,57
13	92,67 (6,50)	97,33 (2,00)	92,67 (6,50)	92,67 (6,50)	97,33 (2,00)	92,67 (6,50)	92,67 (6,50)	97,33 (2,00)	92,67 (6,50)	49,33
14	91,64 (6,00)	94,56 (2,50)	91,64 (6,00)	90,28 (9,00)	95,82 (1,00)	91,64 (6,00)	91,64 (6,00)	94,56 (2,50)	91,64 (6,00)	49,85
15	70,49 (2,50)	55,74 (6,50)	55,74 (6,50)	72,13 (1,00)	47,54 (9,00)	55,74 (6,50)	70,49 (2,50)	55,74 (6,50)	62,30 (4,00)	54,10
16	56,16 (8,00)	68,49 (2,00)	61,64 (6,00)	64,38 (4,50)	69,86 (1,00)	54,79 (9,00)	65,75 (3,00)	57,53 (7,00)	64,38 (4,50)	26,03
17	68,55 (8,00)	75,26 (4,00)	79,08 (2,00)	74,34 (5,00)	73,68 (6,00)	79,34 (1,00)	68,03 (9,00)	70,66 (7,00)	78,16 (3,00)	51,45
18	85,14 (3,50)	71,33 (8,00)	85,14 (3,50)	85,14 (3,50)	71,33 (8,00)	85,14 (3,50)	85,14 (3,50)	71,33 (8,00)	85,14 (3,50)	53,91
19	93,33 (7,50)	100,00 (2,50)	100,00 (2,50)	86,67 (9,00)	96,67 (5,50)	100,00 (2,50)	93,33 (7,50)	96,67 (5,50)	100,00 (2,50)	40,00
20	78,51 (2,00)	71,90 (7,50)	71,90 (7,50)	78,51 (2,00)	78,51 (2,00)	76,45 (6,00)	78,10 (4,00)	77,27 (5,00)	68,18 (9,00)	18,18
21	86,02 (4,50)	53,24 (8,50)	86,02 (4,50)	86,02 (4,50)	53,24 (8,50)	86,02 (4,50)	86,02 (4,50)	86,19 (1,00)	86,02 (4,50)	42,93
22	87,20 (3,00)	91,20 (1,00)	86,24 (4,00)	85,28 (7,00)	85,76 (6,00)	86,08 (5,00)	82,56 (9,00)	82,88 (8,00)	88,64 (2,00)	5,28
23	80,30 (1,00)	70,95 (4,00)	65,93 (9,00)	80,00 (2,00)	66,53 (8,00)	67,44 (6,00)	78,09 (3,00)	67,74 (5,00)	67,04 (7,00)	17,39
24	87,00 (7,00)	91,67 (5,00)	96,67 (1,00)	94,33 (2,00)	93,33 (3,00)	93,00 (4,00)	81,33 (9,00)	86,33 (8,00)	88,33 (6,00)	16,67
25	100,00 (2,50)	98,00 (8,00)	99,00 (6,00)	100,00 (2,50)	100,00 (2,50)	99,00 (6,00)	96,00 (9,00)	99,00 (6,00)	100,00 (2,50)	19,00
26	96,75 (3,50)	94,56 (8,00)	96,75 (3,50)	96,75 (3,50)	94,56 (8,00)	96,75 (3,50)	96,75 (3,50)	94,56 (8,00)	96,75 (3,50)	49,96
27	72,93 (9,00)	82,15 (6,00)	83,55 (4,00)	78,33 (7,00)	86,35 (1,00)	86,23 (2,00)	73,03 (8,00)	82,58 (5,00)	83,58 (3,00)	25,88
28	39,78 (6,00)	60,44 (1,00)	48,57 (3,00)	44,84 (5,00)	54,73 (2,00)	35,16 (8,00)	20,00 (9,00)	48,13 (4,00)	39,34 (7,00)	12,53
29	40,51 (8,00)	53,08 (2,00)	51,03 (3,00)	44,36 (6,00)	55,38 (1,00)	46,92 (4,00)	29,23 (9,00)	42,31 (7,00)	45,64 (5,00)	6,67
30	46,15 (8,00)	60,51 (2,00)	62,56 (1,00)	51,54 (7,00)	55,64 (4,00)	59,74 (3,00)	41,54 (9,00)	54,36 (5,00)	53,33 (6,00)	7,44
31	39,23 (8,00)	53,08 (2,00)	41,28 (7,00)	42,05 (6,00)	53,59 (1,00)	52,56 (3,00)	37,44 (9,00)	47,69 (4,00)	45,38 (5,00)	6,15
32	47,65 (5,00)	54,08 (1,00)	50,31 (2,00)	49,37 (3,00)	48,12 (4,00)	41,07 (8,00)	43,10 (7,00)	38,71 (9,00)	47,34 (6,00)	21,94
acc média	75,91	78,25	78,65	77,53	76,88	76,58	72,91	75,21	76,19	
rank médio	5,27	4,11	4,19	4,63	4,31	5,11	6,53	5,80	5,06	
1 vs todos	6	7	8	8	10	7	4	4	6	

Tabela 14 – Resultados dos experimentos realizados utilizando o NNGE como algoritmo base. Os valores são apresentados no formato *acc (rank)*, no qual *acc* corresponde ao valor da acurácia (%) de classificação sobre o conjunto de teste, e *rank* é a posição relativa do algoritmo em relação aos demais para um dado conjunto de dados. As células destacadas representam os melhores valores de acurácia para cada conjunto de dados.

#	TS+CFS			TS+CS			TS+FCBF			baseline
	lg	2k	Pr	lg	2k	Pr	lg	2k	Pr	
1	80,31 (4,00)	88,49 (2,00)	82,35 (3,00)	90,03 (1,00)	78,77 (5,00)	50,90 (9,00)	60,61 (8,00)	70,33 (7,00)	71,61 (6,00)	2,05
2	63,33 (2,50)	53,33 (5,00)	40,00 (9,00)	63,33 (2,50)	46,67 (8,00)	50,00 (6,50)	66,67 (1,00)	50,00 (6,50)	60,00 (4,00)	20,00
3	85,11 (7,00)	91,78 (2,50)	91,11 (4,50)	85,11 (7,00)	91,78 (2,50)	91,11 (4,50)	85,11 (7,00)	91,89 (1,00)	82,22 (9,00)	33,11
4	46,41 (9,00)	95,08 (2,00)	93,26 (4,00)	94,69 (3,00)	95,44 (1,00)	92,37 (6,00)	89,74 (7,00)	87,68 (8,00)	92,40 (5,00)	53,26
5	96,43 (5,50)	100,00 (1,50)	96,43 (5,50)	96,43 (5,50)	100,00 (1,50)	96,43 (5,50)	96,43 (5,50)	92,86 (9,00)	96,43 (5,50)	53,57
6	77,78 (4,50)	70,59 (6,50)	97,06 (1,50)	77,78 (4,50)	70,59 (6,50)	97,06 (1,50)	59,15 (8,50)	59,15 (8,50)	86,27 (3,00)	30,07
7	85,00 (3,50)	81,00 (9,00)	86,00 (1,50)	83,00 (6,00)	82,00 (7,50)	86,00 (1,50)	85,00 (3,50)	82,00 (7,50)	84,00 (5,00)	64,00
8	98,49 (3,50)	97,33 (7,50)	98,49 (3,50)	98,49 (3,50)	97,33 (7,50)	98,49 (3,50)	98,49 (3,50)	96,86 (9,00)	98,49 (3,50)	49,71
9	56,09 (8,00)	68,46 (3,00)	67,75 (5,50)	70,36 (1,00)	68,40 (4,00)	70,30 (2,00)	48,40 (9,00)	64,02 (7,00)	67,75 (5,50)	4,26
10	65,91 (7,00)	84,09 (4,00)	90,91 (1,50)	56,82 (9,00)	82,95 (5,00)	90,91 (1,50)	65,91 (7,00)	86,36 (3,00)	65,91 (7,00)	15,91
11	70,63 (4,00)	76,10 (3,00)	67,76 (6,00)	77,12 (1,00)	66,34 (7,00)	67,85 (5,00)	50,59 (9,00)	76,59 (2,00)	63,95 (8,00)	14,34
12	92,57 (2,00)	92,00 (4,00)	92,57 (2,00)	89,71 (7,50)	92,57 (2,00)	89,71 (7,50)	84,57 (9,00)	90,86 (6,00)	91,43 (5,00)	12,57
13	92,67 (4,50)	83,33 (8,50)	92,67 (4,50)	92,67 (4,50)	97,33 (1,00)	92,67 (4,50)	92,67 (4,50)	83,33 (8,50)	92,67 (4,50)	49,33
14	94,27 (4,50)	95,63 (2,50)	91,64 (7,00)	91,16 (9,00)	96,40 (1,00)	91,64 (7,00)	94,27 (4,50)	95,63 (2,50)	91,64 (7,00)	49,85
15	59,02 (2,00)	52,46 (8,00)	55,74 (5,00)	68,85 (1,00)	47,54 (9,00)	55,74 (5,00)	54,10 (7,00)	57,38 (3,00)	55,74 (5,00)	54,10
16	64,38 (5,50)	65,75 (3,00)	76,71 (1,00)	64,38 (5,50)	65,75 (3,00)	58,90 (8,00)	58,90 (8,00)	65,75 (3,00)	58,90 (8,00)	26,03
17	68,55 (8,00)	75,79 (4,00)	79,34 (1,00)	75,92 (3,00)	75,26 (5,00)	76,32 (2,00)	68,03 (9,00)	71,18 (7,00)	71,32 (6,00)	51,45
18	85,14 (3,50)	72,76 (9,00)	85,14 (3,50)	85,14 (3,50)	80,99 (7,00)	85,14 (3,50)	85,14 (3,50)	80,19 (8,00)	85,14 (3,50)	53,91
19	93,33 (7,00)	100,00 (3,00)	100,00 (3,00)	86,67 (8,50)	96,67 (6,00)	100,00 (3,00)	86,67 (8,50)	100,00 (3,00)	100,00 (3,00)	40,00
20	69,01 (9,00)	75,62 (3,00)	75,21 (4,00)	73,55 (5,50)	76,86 (1,50)	70,66 (8,00)	76,86 (1,50)	73,55 (5,50)	72,31 (7,00)	18,18
21	86,02 (4,50)	53,24 (8,50)	86,02 (4,50)	86,02 (4,50)	53,24 (8,50)	86,02 (4,50)	86,02 (4,50)	86,19 (1,00)	86,02 (4,50)	42,93
22	84,00 (8,00)	91,36 (1,00)	86,88 (3,00)	84,32 (7,00)	86,08 (4,00)	85,76 (5,00)	69,60 (9,00)	84,96 (6,00)	88,48 (2,00)	5,28
23	82,31 (2,00)	70,95 (4,00)	67,94 (6,00)	83,12 (1,00)	66,53 (8,00)	69,45 (5,00)	76,98 (3,00)	66,93 (7,00)	56,98 (9,00)	17,39
24	85,67 (7,00)	91,33 (5,00)	96,67 (1,00)	95,67 (2,00)	93,33 (4,00)	94,33 (3,00)	82,33 (9,00)	86,33 (6,00)	84,67 (8,00)	16,67
25	100,00 (2,00)	99,00 (4,50)	95,00 (9,00)	100,00 (2,00)	100,00 (2,00)	99,00 (4,50)	96,00 (8,00)	98,00 (6,50)	98,00 (6,50)	19,00
26	96,75 (3,50)	94,56 (8,00)	96,75 (3,50)	96,75 (3,50)	94,56 (8,00)	96,75 (3,50)	96,75 (3,50)	94,56 (8,00)	96,75 (3,50)	49,96
27	72,93 (9,00)	79,18 (6,00)	83,65 (5,00)	76,05 (7,00)	86,33 (2,00)	86,68 (1,00)	72,95 (8,00)	84,05 (3,00)	83,73 (4,00)	25,88
28	31,65 (8,00)	61,98 (1,00)	44,62 (4,00)	39,78 (6,00)	55,38 (2,00)	35,60 (7,00)	14,73 (9,00)	52,31 (3,00)	42,42 (5,00)	12,53
29	39,74 (8,00)	56,67 (1,00)	55,64 (2,00)	45,13 (7,00)	52,05 (5,00)	52,82 (4,00)	30,26 (9,00)	50,51 (6,00)	53,08 (3,00)	6,67
30	46,15 (8,00)	66,15 (2,00)	67,95 (1,00)	50,00 (7,00)	53,08 (5,00)	59,49 (4,00)	42,05 (9,00)	52,05 (6,00)	60,26 (3,00)	7,44
31	35,64 (8,00)	57,18 (1,00)	56,67 (3,00)	35,90 (7,00)	56,41 (4,00)	55,38 (5,00)	33,33 (9,00)	56,92 (2,00)	44,10 (6,00)	6,15
32	47,49 (4,00)	60,19 (1,00)	44,67 (6,00)	48,90 (3,00)	50,78 (2,00)	40,91 (9,00)	41,69 (8,00)	44,20 (7,00)	44,83 (5,00)	21,94
acc média	73,52	78,17	79,46	76,96	76,79	77,32	70,31	76,02	75,86	
rank médio	5,52	4,19	3,89	4,66	4,55	4,70	6,67	5,52	5,31	
1 vs todos	5	7	12	9	7	9	6	3	4	



Tabela 15 – Resultados dos experimentos realizados utilizando o HID como algoritmo base. Os valores são apresentados no formato *acc (rank)*, no qual *acc* corresponde ao valor da acurácia (%) de classificação sobre o conjunto de teste, e *rank* é a posição relativa do algoritmo em relação aos demais para um dado conjunto de dados. As células destacadas representam os melhores valores de acurácia para cada conjunto de dados.

#	TS+CFS			TS+CS			TS+FCBF			baseline
	lg	2k	Pr	lg	2k	Pr	lg	2k	Pr	
1	80,56 (4,00)	85,17 (3,00)	89,77 (1,00)	88,49 (2,00)	73,91 (6,00)	51,92 (9,00)	61,89 (8,00)	67,52 (7,00)	74,17 (5,00)	2,05
2	70,00 (3,00)	66,67 (4,50)	76,67 (1,00)	66,67 (4,50)	56,67 (6,00)	53,33 (7,50)	73,33 (2,00)	53,33 (7,50)	46,67 (9,00)	20,00
3	85,11 (7,00)	91,78 (2,50)	91,11 (4,50)	85,11 (7,00)	91,78 (2,50)	91,11 (4,50)	85,11 (7,00)	91,89 (1,00)	82,22 (9,00)	33,11
4	90,78 (8,00)	95,81 (1,00)	92,73 (6,00)	94,38 (4,00)	95,00 (2,00)	91,41 (7,00)	89,74 (9,00)	94,77 (3,00)	94,09 (5,00)	53,26
5	96,43 (5,50)	100,00 (1,50)	96,43 (5,50)	96,43 (5,50)	100,00 (1,50)	96,43 (5,50)	96,43 (5,50)	92,86 (9,00)	96,43 (5,50)	53,57
6	77,78 (4,50)	70,59 (6,50)	88,24 (1,50)	77,78 (4,50)	70,59 (6,50)	88,24 (1,50)	59,15 (8,50)	59,15 (8,50)	78,43 (3,00)	30,07
7	83,00 (5,00)	82,00 (8,00)	86,00 (2,00)	83,00 (5,00)	82,00 (8,00)	86,00 (2,00)	83,00 (5,00)	82,00 (8,00)	86,00 (2,00)	64,00
8	98,49 (3,50)	97,33 (7,50)	98,49 (3,50)	98,49 (3,50)	97,33 (7,50)	98,49 (3,50)	98,49 (3,50)	96,86 (9,00)	98,49 (3,50)	49,71
9	66,98 (7,00)	74,79 (1,00)	74,67 (2,00)	70,83 (5,00)	65,44 (8,00)	68,11 (6,00)	58,70 (9,00)	73,85 (4,00)	73,96 (3,00)	4,26
10	56,82 (8,50)	82,95 (3,50)	90,91 (1,00)	56,82 (8,50)	82,95 (3,50)	85,23 (2,00)	68,18 (6,00)	81,82 (5,00)	65,91 (7,00)	15,91
11	75,27 (5,00)	83,56 (1,00)	72,78 (6,00)	78,05 (3,00)	66,78 (8,00)	71,12 (7,00)	57,66 (9,00)	78,93 (2,00)	75,41 (4,00)	14,34
12	91,43 (5,50)	96,00 (1,00)	93,71 (3,00)	91,43 (5,50)	94,29 (2,00)	89,14 (8,00)	85,14 (9,00)	89,71 (7,00)	93,14 (4,00)	12,57
13	92,67 (6,50)	96,67 (2,00)	92,67 (6,50)	92,67 (6,50)	97,33 (1,00)	92,67 (6,50)	92,67 (6,50)	94,67 (3,00)	92,67 (6,50)	49,33
14	90,28 (8,50)	96,60 (1,50)	94,17 (5,00)	90,28 (8,50)	95,82 (3,00)	91,64 (7,00)	94,66 (4,00)	96,60 (1,50)	92,03 (6,00)	49,85
15	68,85 (3,00)	54,10 (8,00)	62,30 (4,50)	72,13 (2,00)	45,90 (9,00)	57,38 (6,00)	73,77 (1,00)	55,74 (7,00)	62,30 (4,50)	54,10
16	56,16 (8,50)	69,86 (2,00)	61,64 (5,50)	68,49 (3,00)	71,23 (1,00)	61,64 (5,50)	56,16 (8,50)	60,27 (7,00)	63,01 (4,00)	26,03
17	69,61 (8,00)	80,79 (3,00)	81,71 (1,00)	75,66 (5,00)	74,08 (6,00)	81,05 (2,00)	68,95 (9,00)	71,05 (7,00)	76,18 (4,00)	51,45
18	85,14 (4,50)	83,87 (8,00)	85,14 (4,50)	85,14 (4,50)	71,33 (9,00)	85,14 (4,50)	85,14 (4,50)	87,46 (1,00)	85,14 (4,50)	53,91
19	86,67 (7,50)	96,67 (5,00)	100,00 (2,00)	86,67 (7,50)	96,67 (5,00)	100,00 (2,00)	83,33 (9,00)	96,67 (5,00)	100,00 (2,00)	40,00
20	79,34 (3,00)	83,88 (1,00)	78,10 (4,50)	75,21 (8,50)	77,69 (6,00)	75,21 (8,50)	78,10 (4,50)	75,62 (7,00)	81,41 (2,00)	18,18
21	86,02 (4,50)	53,24 (8,50)	86,02 (4,50)	86,02 (4,50)	53,24 (8,50)	86,02 (4,50)	86,02 (4,50)	86,19 (1,00)	86,02 (4,50)	42,93
22	89,44 (4,00)	94,56 (1,00)	91,36 (2,00)	88,32 (6,00)	88,80 (5,00)	87,68 (7,00)	84,00 (8,00)	83,84 (9,00)	90,24 (3,00)	5,28
23	81,01 (1,00)	54,27 (8,00)	67,94 (6,00)	80,00 (2,00)	71,26 (4,00)	69,45 (5,00)	76,98 (3,00)	50,05 (9,00)	57,19 (7,00)	17,39
24	82,00 (8,00)	90,00 (5,00)	92,00 (4,00)	94,00 (1,00)	92,67 (2,50)	92,67 (2,50)	80,33 (9,00)	84,00 (7,00)	87,67 (6,00)	16,67
25	100,00 (2,50)	100,00 (2,50)	99,00 (5,50)	100,00 (2,50)	100,00 (2,50)	98,00 (7,50)	96,00 (9,00)	99,00 (5,50)	98,00 (7,50)	19,00
26	96,75 (3,50)	94,56 (8,00)	96,75 (3,50)	96,75 (3,50)	94,56 (8,00)	96,75 (3,50)	96,75 (3,50)	94,56 (8,00)	96,75 (3,50)	49,96
27	72,70 (9,00)	84,50 (3,00)	83,80 (5,00)	79,28 (7,00)	87,35 (1,00)	86,63 (2,00)	72,98 (8,00)	82,93 (6,00)	83,83 (4,00)	25,88
28	38,90 (7,50)	68,13 (1,00)	46,59 (4,00)	46,15 (5,00)	56,48 (2,00)	38,90 (7,50)	21,76 (9,00)	49,67 (3,00)	40,88 (6,00)	12,53
29	41,03 (8,00)	65,64 (2,00)	68,46 (1,00)	42,82 (7,00)	53,59 (5,00)	57,18 (4,00)	32,31 (9,00)	60,51 (3,00)	51,54 (6,00)	6,67
30	45,13 (8,00)	72,05 (2,00)	76,92 (1,00)	51,79 (7,00)	55,13 (6,00)	64,62 (3,00)	44,10 (9,00)	61,79 (4,00)	58,46 (5,00)	7,44
31	39,49 (8,00)	61,54 (2,00)	68,46 (1,00)	42,05 (7,00)	59,49 (4,00)	58,21 (5,00)	37,69 (9,00)	61,28 (3,00)	56,41 (6,00)	6,15
32	47,96 (6,00)	66,61 (1,00)	59,40 (2,00)	50,31 (3,00)	48,59 (5,00)	47,02 (7,00)	42,63 (9,00)	44,51 (8,00)	49,69 (4,00)	21,94
acc média	75,68	81,07	82,62	77,85	77,12	78,07	72,54	76,85	77,32	
rank médio	5,81	3,61	3,44	4,97	4,84	5,13	6,83	5,50	4,88	
1 vs todos	4	11	12	4	5	5	3	4	4	

**Análise do algoritmo base J48:** Por meio da aplicação do teste de Friedman sobre os valores de *rank* médio descritos na tabela 13, a hipótese nula de que todos os modelos construídos com o algoritmo base J48 são equivalentes, em termos de acurácia, foi rejeitada com valores de  $F_F = 2,96$  e  $p\text{-valor} = 0,0035$ . As diferenças verificadas no diagrama de diferença crítica são apresentadas na figura 36.

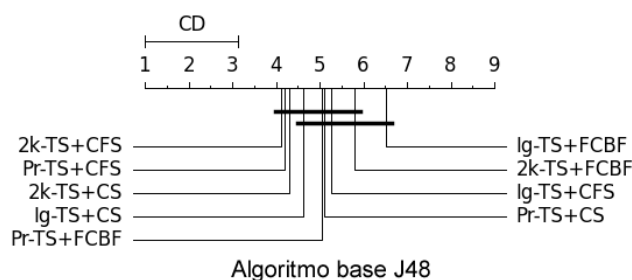


Figura 36 – Diagrama de diferença crítica para o pós-teste de Nemenyi, análise do algoritmo base J48.

Com base na análise dos resultados representados na figura 36, pode-se verificar que nenhum algoritmo de aprendizado de limiares foi estatisticamente superior. De modo semelhante, nenhuma das variações da abordagem reduzida apresentou superioridade estatística em relação aos demais. A única diferença significativa utilizando o algoritmo base J48 foi verificada entre os modelos gerados pelas estratégias 2k-TS+CFS e Ig-TS+FCBF.

Dentre os algoritmos de binarização, os melhores resultados, em termos de *rank* médio, foram reportados pela estratégia 2k, considerando as abordagens TS+CFS e TS+CS, e Pr, considerando a abordagem TS+FCBF. Adicionalmente, o melhor resultado geral foi verificado para a estratégia de combinação 2k-TS+CFS.

Ressalta-se que, para nenhuma abordagem de representação binária, utilizando como base o algoritmo J48, o desempenho em termos de acurácia foi inferior ao *baseline*.

**Análise do algoritmo base NNGE:** Por meio da aplicação do teste de Friedman sobre os valores de *rank* médio descritos na tabela 14, a hipótese nula de que todos os modelos construídos com o algoritmo base NNGE são equivalentes, em termos de acurácia, foi rejeitada com valores de  $F_F = 3,33$  e  $p\text{-valor} = 0,0012$ . As diferenças verificadas no diagrama de diferença crítica são apresentadas na figura 37.

Com base na análise dos resultados representados na figura 37, pode-se verificar que nenhum algoritmo de binarização foi estatisticamente superior. Também, nenhuma das variações da abordagem reduzida apresentou superioridade estatística em relação aos demais. A única diferença significativa utilizando o algoritmo base NNGE foi verificada entre os modelos gerados pelas estratégias Pr-TS+CFS e Ig-TS+FCBF.

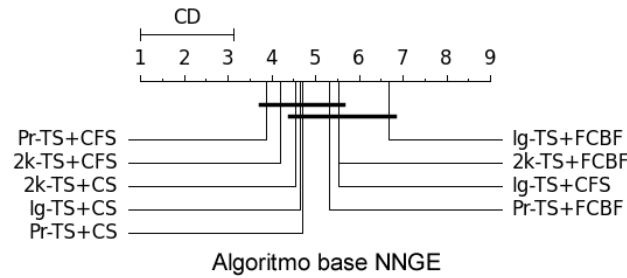


Figura 37 – Diagrama de diferença crítica para o pós-teste de Nemenyi, análise do algoritmo base NNGE.

Dentre os algoritmos de binarização, os melhores resultados, em termos de *rank* médio, foram reportados pelas estratégias Pr, 2k e lg, considerando as abordagens TS+CFS, TS+CS e TS+FCBF, respectivamente. Em relação às variações da abordagem reduzida, todas as combinações baseadas em TS+CS foram melhores, em termos de *rank* médio, que as baseadas em TS+FCBF. Adicionalmente, o melhor resultado geral foi verificado para a estratégia Pr-TS+CFS.

Ressalta-se que, para nenhuma abordagem de representação binária, utilizando como base o algoritmo NNGE, o desempenho em termos de acurácia foi inferior ao *baseline*.

**Análise do algoritmo base HID:** Por meio da aplicação do teste de Friedman sobre os valores de *rank* médio descritos na tabela 15, a hipótese nula de que todos os modelos construídos com o algoritmo base HID são equivalentes, em termos de acurácia, foi rejeitada com valores de  $F_F = 5,31$  e  $p\text{-valor} \leq 0,001$ . As diferenças verificadas no diagrama de diferença crítica são apresentadas na figura 38.

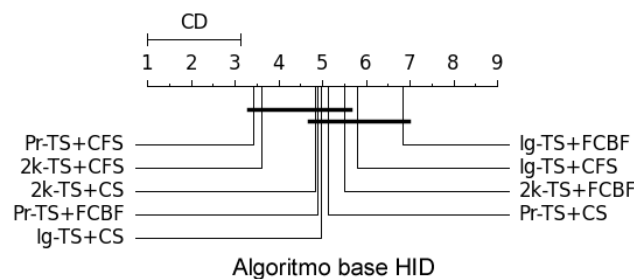


Figura 38 – Diagrama de diferença crítica para o pós-teste de Nemenyi, análise do algoritmo base HID.

Pela análise dos resultados representados na figura 38, pode-se verificar, por meio do pós-teste de Nemenyi, que os algoritmos de aprendizado de limiares Pr e 2k foram estatisticamente superiores ao lg, considerando a abordagem TS+CFS. Nenhuma das variações da abordagem reduzida apresentou superioridade estatística em relação aos demais.

Dentre os algoritmos de aprendizado de limiares, os melhores resultados, em termos de *rank* médio, foram reportados pela estratégia Pr, considerando as abordagens TS+CFS e TS+FCBF, e 2k, considerando a abordagem TS+FCBF. Adicionalmente, o melhor resultado geral foi verificado para a estratégia Pr-TS+CFS.

Ressalta-se que, para nenhuma abordagem de representação binária, utilizando como base o algoritmo HID, o desempenho em termos de acurácia foi inferior ao *baseline*.

Com base nas análises dos resultados reportados nesta avaliação, sobre os 32 conjuntos de dados utilizados, verificou-se que somente para o algoritmo HID, considerando a abordagem TS+CFS, foram verificadas diferenças significativas entre as estratégias de aprendizado de limiares. Especificamente, as estratégias Pr e 2k foram superiores à lg. Em relação às abordagens da transformada *shapenet* reduzida, não houve alguma que apresentasse superioridade estatística sobre as demais. Dado esse cenário, com exceção das diferenças identificadas para o algoritmo HID, pode-se concluir que, para os conjuntos dados analisados, qualquer algoritmo de aprendizado ou qualquer variação da transformada *shapenet* reduzida, pode ser utilizado para a representação binária. Ressalta-se que as avaliações foram realizadas com base nos algoritmos de aprendizado de máquina utilizados, e que portanto os resultados estão restritos a análise exclusiva sobre esses algoritmos.

#### 5.6.6 Avaliação 3 – Resultados e discussão

Nessa avaliação foram comparados os desempenhos dos algoritmos base J48, NNGE e HID. Para isso foram utilizadas as estratégias de combinação que apresentaram os melhores resultados na Avaliação 2, em termos de valores de *rank* médio. Nesse cenário, foram avaliados o algoritmo J48 com a estratégia 2k-TS+CFS e os algoritmos NNGE e HID utilizando a estratégia Pr-TS+CFS. Os resultados de acurácia de cada um dos modelos construídos são apresentados nas tabelas 36 (J48 usando 2k-TS+CFS), 37 (NNGE usando Pr-TS+CFS) e 38 (HID usando Pr-TS+CFS).

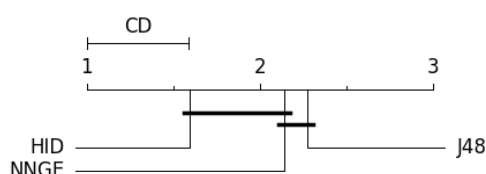


Figura 39 – Diagrama de diferença crítica para o pós-teste de Nemenyi - Avaliação 3

Por meio da aplicação do teste de Friedman, a hipótese nula de que todos os três modelos construídos são equivalentes, em termos de acurácia, foi rejeitada com valores de  $F_F = 4,64$  e  $p\text{-valor} = 0,0132$ . Para verificar quais são as diferenças

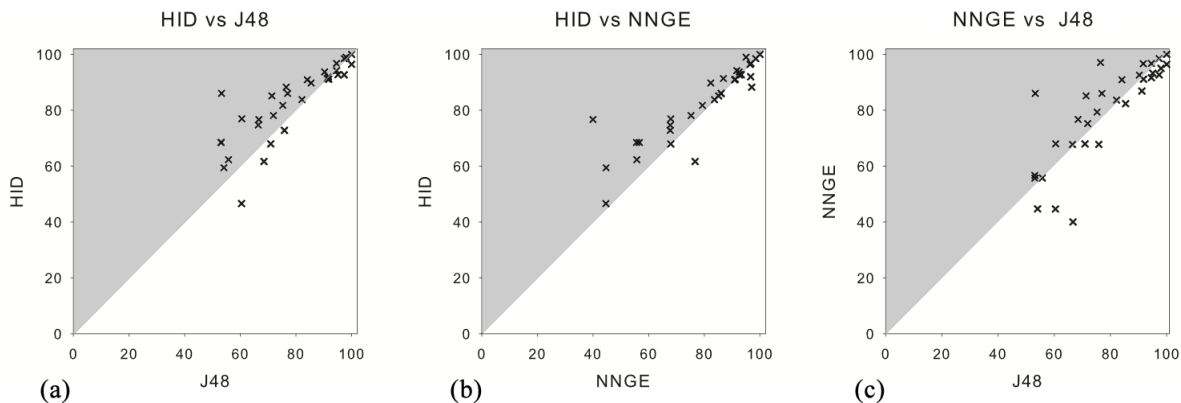


Figura 40 – Comparação entre os algoritmos (a) HID e J48, (b) HID e NNGE, e (c) NNGE e J48, em termos de acurácia no conjunto de teste.

significativas, o pós-teste de Nemenyi foi aplicado com  $\alpha = 5\%$ , o qual apontou valor de diferença crítica de  $CD = 0,58$ . Os valores de *rank* médio foram 2,27, 2,14 e 1,59 para os algoritmos J48, HID e NNGE, respectivamente.

Pela análise dos resultados representados na figura 39, pode-se verificar que o algoritmo HID foi estatisticamente superior em relação ao J48. Em relação ao algoritmo NNGE nenhuma diferença significativa foi verificada.

Sobre os 32 conjuntos de dados utilizados nesta avaliação, o algoritmo HID apresentou acurácia superior ao J48 em 22 conjuntos, enquanto foi inferior em 9 conjuntos, e empatou apenas no conjunto *OliveOil* (#19). As maiores diferenças foram verificadas no conjunto *SonyAIBORobotSurface* (#25) que passou de 53,24% (J48) para 86,02% (HID), diferença de 32,78%; e no conjunto *50words* (#28) que passou de 60,44% (J48) para 46,59% (HID), diferença de 13,85%. Em relação ao algoritmo NNGE, HID apresentou os melhores resultados para 17 conjuntos, enquanto foi pior para 4 conjuntos, e empatou em 11 conjuntos. As maiores diferenças foram observadas no conjunto *Beef* (#3) que passou de 40,00% (NNGE) para 76,67% (HID), diferença de 36,67%; e no conjunto *Lighting7* (#20) que passou de 76,71% (J48) para 61,64% (HID), diferença de 15,07%. Na comparação entre os algoritmos J48 e NNGE, em 2 conjuntos foram verificados empates e em 17 conjuntos o algoritmo NNGE apresentou acurácia superior ao J48. As maiores diferenças foram verificadas no conjunto no *Beef* (#3) que passou de 40,00% (NNGE) para 66,67% (J48), diferença de 26,67%; e no conjunto *SonyAIBORobotSurface* (#25) que passou de 53,24% (J48) para 86,02% (NNGE), diferença de 32,78%.

Na figura 40 são apresentados os resultados comparativos, na qual os pontos localizados acima da linha diagonal (região em cor cinza) de cada gráfico representam os conjuntos de dados que apresentaram melhor acurácia para os algoritmos HID (figura (a) e figura (b)) e NNGE (figura (c)).

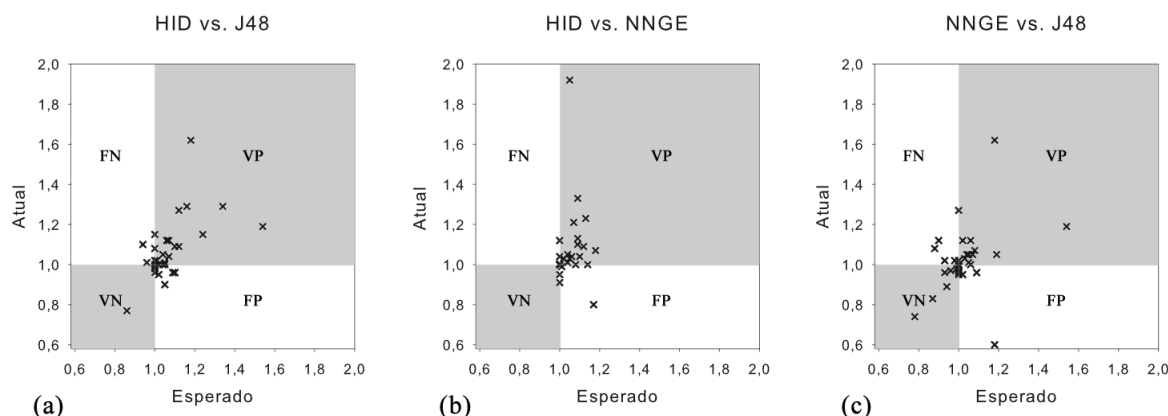


Figura 41 – Comparação entre os algoritmos (a) HID e J48, (b) HID e NNGE, e (c) NNGE e J48, utilizando a análise *Texas Sharpshooter Fallacy*.

	VP	VN	FP	FN
HID vs. J48	17	1	5	2
HID vs. NNGE	15	1	2	0
NNGE vs. HID	11	6	4	4

Tabela 16 – Sumário dos resultados da análise *Texas Sharpshooter Fallacy*.

Assim como realizado nas avaliações do capítulo anterior (seção 4.3.6), na figura 41 são apresentados os gráficos da análise *Texas Sharpshooter Fallacy* construídos para comparar o algoritmo HID com J48 e NNGE (figura (a) e figura (b)), e NNGE com J48 (figura (c)).

Na tabela 16 estão descritos os resultados registrados para cada uma das quatro regiões de interesse do gráfico (VP, VN, FP, FN). Com base nesses dados, é possível observar que o algoritmo HID teve seu desempenho estimado corretamente para 72,00% e 88,89% dos conjuntos de dados, quando em comparação com os algoritmos J48 e NNGE, respectivamente. O desempenho do algoritmo NNGE foi estimado corretamente para 68% dos conjuntos de dados, quando em comparação com o J48.

Os resultados acima descritos, evidenciam a capacidade do algoritmo HID em possibilitar a estimativa correta de seus desempenhos para a maioria dos conjuntos de dados avaliados, quando comparados aos algoritmos J48 e NNGE.

Considerando a representação binária proposta e a análise dos resultados reportados para a Avaliação 3, é possível verificar que as abordagens baseadas na construção de modelos simbólicos híbridos apresentaram desempenho superior em relação ao algoritmo J48. No entanto, somente foi verificada diferença estatística entre os algoritmos HID e J48. Nesse aspecto é importante ressaltar que a construção da árvore de decisão por meio do algoritmo HID não descarta quaisquer atributos da representação binária provida pelo algoritmo BST.

Em relação aos métodos híbridos, apesar de nenhuma diferença significativa ter sido constatada em relação ao algoritmo NNGE, os resultados de acurácia dos modelos sugerem um desempenho superior do algoritmo HID. Além desse aspecto, ressalta-se que o algoritmo HID possibilita contornar as principais desvantagens do algoritmo NNGE. Como não existe a necessidade de definição de hiper-retângulos, o algoritmo HID evita a sobre-generalização, pois sempre associa um novo exemplo ao mais próximo existente no conjunto de treinamento. Por meio da utilização dos diagramas algébricos de decisão é possível compartilhar partes da estrutura da árvore de decisão, de modo que regras redundantes podem ser eliminadas. E ao contrário do NNGE, no qual as regras não são ordenadas, no algoritmo HID existe uma ordenação única dos testes a serem executados, o que facilita a inteligibilidade dos modelos.

#### 5.6.7 Avaliação 4 – Resultados e discussão

Nesta avaliação o algoritmo HID foi analisado em relação aos algoritmos baseados na primitiva *shapelet* por meio da abordagem embutida (LS e FS) e da abordagem transformada (TS+CS, TS+CFS e TS+FCBF). Adicionalmente, o algoritmo HID também foi comparado em relação ao algoritmo 1NN, usando as medidas de distância ED e DTW. Para isso, o algoritmo HID foi avaliado utilizando a representação binária provida pela estratégia Pr-TS+CFS, a qual apresentou o melhor *rank* médio na Avaliação 2. Nesta avaliação foram utilizados 27 conjuntos de dados, apresentados na tabela 2 (capítulo 4), que são os mesmos conjuntos avaliados nos trabalhos dos quais os resultados foram coletados. Ressalta-se que os resultados reportados para os algoritmos LS e FS foram extraídos de [Rakthanmanon e Keogh \(2013\)](#); no qual os experimentos foram conduzidos com parâmetros de tamanho de subsequência  $min = 10$  e  $max = m - 10$ . Os resultados para os algoritmos de 1NN ED e 1NN DTW foram retirados do trabalho de [Silva, Souza e Batista \(2013\)](#).

Os resultados desta avaliação estão descritos na tabela 17. Os valores são apresentados no formato *acc (rank)*, no qual *acc* corresponde ao valor da acurácia de classificação sobre o conjunto de teste (em termos de porcentagem), e *rank* é a posição relativa do algoritmo em relação aos demais para um dado conjunto de dados. As últimas três linhas representam, para cada algoritmo, os valores de acurácia média, *rank* médio, e quantidade de vezes que apresentou a melhor acurácia em um conjunto de dados (1 vs. todos), respectivamente. As células destacadas representam os melhores valores de acurácia para um determinado conjunto de dados, considerando todos os algoritmos avaliados.

Tabela 17 – Resultados dos experimentos realizados utilizando o HID em comparação com os algoritmos LS, FS, 1NN-ED, 1NN-DTW, TS+CFS, TS+CS e TS+FCBF. Os valores são apresentados no formato *acc (rank)*, no qual *acc* corresponde ao valor da acurácia (%) de classificação sobre o conjunto de teste, e *rank* é a posição relativa do algoritmo em relação aos demais para um dado conjunto de dados. As células destacadas representam os melhores valores de acurácia para cada conjunto de dados.

#	LS	FS	1NN-ED	1NN-DTW	TS+CFS	TS+CS	TS+FCBF	HID
1	58,60 (7,00)	48,60 (8,00)	61,10 (5,00)	60,40 (6,00)	91,56 (3,00)	92,33 (1,00)	91,82 (2,00)	89,77 (4,00)
2	56,70 (5,00)	55,30 (6,00)	53,30 (7,00)	50,00 (8,00)	73,33 (2,00)	63,33 (4,00)	70,00 (3,00)	76,67 (1,00)
3	88,60 (6,00)	94,70 (2,00)	85,20 (7,00)	99,70 (1,00)	81,00 (8,00)	90,78 (5,00)	91,89 (3,00)	91,11 (4,00)
4	61,80 (7,00)	58,30 (8,00)	65,00 (5,00)	64,80 (6,00)	98,33 (2,00)	99,06 (1,00)	98,02 (3,00)	92,73 (4,00)
5	96,40 (4,00)	93,20 (5,00)	75,00 (8,00)	82,10 (7,00)	85,71 (6,00)	96,43 (2,00)	96,43 (2,00)	96,43 (2,00)
6	80,10 (5,00)	88,30 (3,00)	93,50 (2,00)	96,70 (1,00)	67,65 (6,00)	56,21 (7,00)	47,39 (8,00)	88,24 (4,00)
7	87,00 (2,00)	76,60 (7,00)	88,00 (1,00)	77,00 (6,00)	85,00 (4,00)	83,00 (5,00)	73,00 (8,00)	86,00 (3,00)
8	99,40 (2,00)	99,60 (1,00)	79,70 (7,00)	76,80 (8,00)	97,10 (5,00)	97,10 (5,00)	97,10 (5,00)	98,49 (3,00)
9	65,90 (7,00)	58,90 (8,00)	71,40 (4,00)	80,80 (1,00)	67,99 (6,00)	72,78 (3,00)	70,06 (5,00)	74,67 (2,00)
10	48,90 (8,00)	91,00 (1,00)	78,40 (4,00)	83,00 (3,00)	51,14 (7,00)	73,86 (5,50)	73,86 (5,50)	90,91 (2,00)
11	66,20 (8,00)	67,20 (7,00)	76,90 (4,00)	90,49 (1,00)	78,00 (3,00)	69,80 (6,00)	79,85 (2,00)	72,78 (5,00)
12	77,70 (8,00)	80,30 (6,00)	78,30 (7,00)	83,30 (5,00)	94,29 (1,00)	90,29 (4,00)	91,43 (3,00)	93,71 (2,00)
13	89,30 (8,00)	93,90 (1,00)	91,30 (6,00)	90,70 (7,00)	92,00 (4,00)	92,00 (4,00)	92,00 (4,00)	92,67 (2,00)
14	93,60 (4,00)	90,50 (7,00)	95,50 (1,00)	95,00 (2,00)	91,64 (5,50)	90,28 (8,00)	91,64 (5,50)	94,17 (3,00)
15	42,60 (8,00)	70,50 (3,00)	75,40 (2,00)	86,90 (1,00)	65,57 (4,50)	50,82 (7,00)	65,57 (4,50)	62,30 (6,00)
16	54,80 (6,00)	76,70 (1,00)	57,50 (5,00)	72,60 (2,00)	53,42 (7,00)	60,27 (4,00)	52,05 (8,00)	61,64 (3,00)
17	58,70 (7,00)	56,70 (8,00)	68,40 (6,00)	73,70 (5,00)	85,53 (1,50)	81,32 (4,00)	85,53 (1,50)	81,71 (3,00)
18	83,20 (7,00)	78,30 (8,00)	87,90 (1,00)	83,50 (6,00)	85,30 (3,00)	85,30 (3,00)	85,30 (3,00)	85,14 (5,00)
19	83,33 (7,00)	72,30 (8,00)	86,70 (3,50)	86,70 (3,50)	90,00 (2,00)	86,67 (5,50)	86,67 (5,50)	100,00 (1,00)
20	68,60 (4,00)	68,00 (5,00)	51,70 (8,00)	59,10 (7,00)	78,10 (1,50)	64,05 (6,00)	77,27 (3,00)	78,10 (1,50)
21	86,00 (4,00)	68,60 (8,00)	69,50 (7,00)	72,50 (6,00)	73,38 (5,00)	86,02 (2,00)	86,02 (2,00)	86,02 (2,00)
22	81,30 (5,00)	73,10 (8,00)	78,70 (7,00)	79,00 (6,00)	89,60 (3,00)	90,08 (2,00)	88,80 (4,00)	91,36 (1,00)
23	64,30 (8,00)	93,20 (2,00)	90,00 (3,00)	95,00 (1,00)	66,33 (7,00)	75,38 (4,00)	74,17 (5,00)	67,94 (6,00)
24	47,00 (8,00)	91,90 (6,00)	88,00 (7,00)	99,30 (1,00)	97,33 (2,00)	95,00 (4,00)	97,00 (3,00)	92,00 (5,00)
25	100,00 (1,50)	99,80 (3,00)	76,00 (8,00)	100,00 (1,50)	95,00 (7,00)	97,00 (5,50)	97,00 (5,50)	99,00 (4,00)
26	85,60 (7,00)	91,00 (5,50)	91,00 (5,50)	100,00 (1,00)	85,51 (8,00)	96,58 (3,50)	96,58 (3,50)	96,75 (2,00)
27	53,90 (8,00)	88,70 (2,00)	74,70 (7,00)	90,40 (1,00)	85,53 (3,00)	84,18 (4,00)	82,23 (6,00)	83,80 (5,00)
acc média	73,32	78,71	77,34	82,57	81,68	82,22	82,91	86,08
rank médio	5,98	5,09	5,11	3,85	4,33	4,26	4,20	3,17
1 vs todos	1	4	3	10	3	4	3	6



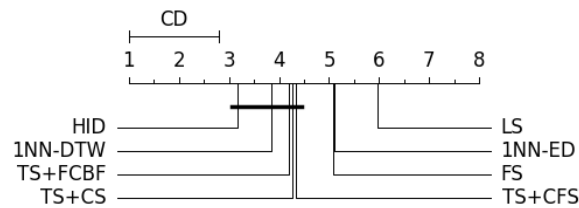


Figura 42 – Diagrama de diferença crítica para o pós-teste de Bonferroni-Dunn - Avaliação 4.

Por meio da aplicação do teste de Friedman com nível de significância de  $\alpha = 5\%$ ,  $K = 8$  e  $N = 27$  sobre os valores de *rank* médio descritos na tabela 17, a hipótese nula de que todos os modelos são equivalentes, em termos de acurácia, foi rejeitada com valores de  $F_F = 3,66$  e  $p\text{-valor} = 0,001$ . Para verificar quais são as diferenças significativas em relação ao algoritmo TS+, o pós-teste de Bonferroni-Dunn foi aplicado com  $\alpha = 5\%$ , o qual apontou valor de diferença crítica de  $CD = 1,79$ .

Pela análise dos resultados representados na figura 42, pode-se verificar que o algoritmo HID foi estatisticamente superior em relação aos algoritmos LS e FS, e em relação ao algoritmo 1NN utilizando a distância ED. Na comparação com as variações da transformada *shapelet* reduzida e com 1NN DTW, nenhuma diferença significativa foi constatada. No entanto, pode-se notar que o algoritmo HID apresentou o melhor valor de *rank* médio, o que demonstra seu desempenho competitivo para a maioria dos conjuntos de dados analisados.

Na figura 43 são apresentados os resultados comparativos, na qual os pontos localizados acima da linha diagonal de cada gráfico (região em cor cinza) representam os conjuntos de dados que apresentaram melhor acurácia para o algoritmo HID. Uma maior acurácia para o algoritmo HID em comparação com LS, FS, 1NN ED, 1NN DTW, TS+CFS, TS+CS e TS+FCBF, foi verificada em 88,89%, 62,96%, 74,07%, 55,56%, 62,96%, 70,37% e 59,26% dos conjuntos avaliados, respectivamente. Empates foram registrados somente em relação aos algoritmos TS+CFS, TS+CS e TS+FCBF, em 3,70%, 7,41% e 7,41% dos conjuntos. Na tabela 18 são apresentados os comparativos em termos de vitórias, empates e derrotas, do algoritmo HID em relação aos demais utilizados nessa avaliação.

Tabela 18 – Resultados da comparação entre o algoritmo HID e os demais algoritmos analisados, em termos de vitórias, derrotas e empates.

	LS	FS	1NN ED	1NN DTW	TS+CFS	TS+CS	TS+FCBF
Vitórias	24	17	20	15	17	19	16
Derrotas	3	10	7	12	9	6	9
Empates	0	0	0	0	1	2	2

Na figura 44 são apresentados os gráficos da análise *Texas Sharpshooter*

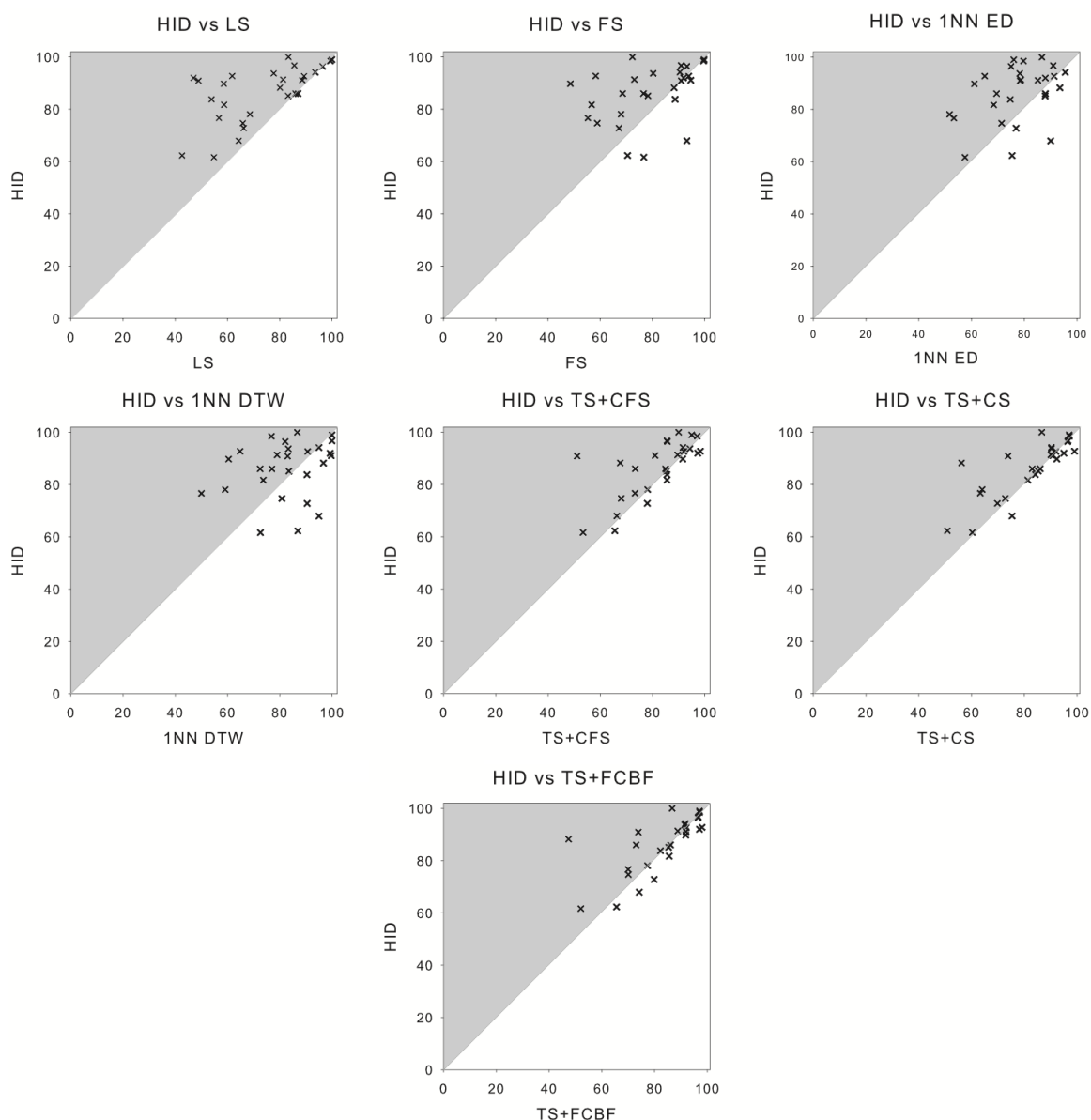


Figura 43 – Comparação entre o algoritmo HID e os algoritmos da abordagem embutida (LS e FS), da abordagem transformada (TS+CS, TS+CFS e TS+FCBF) e do estado da arte (1NN ED e 1NN DTW), em termos de acurácia no conjunto de teste.

*Fallacy* construídos para comparar o algoritmo HID os demais algoritmos utilizados. Na tabela 19 estão descritos os resultados registrados para cada uma das quatro regiões de interesse do gráfico (VP, VN, FP, FN). Com base nesses dados, é possível observar que o algoritmo HID teve seu desempenho estimado corretamente para 73,00%, 80,00%, 39,00%, 73,00% e 82,00% dos conjuntos de dados, quando em comparação com os algoritmos TS+CFS, TS+CS, TS+FCBF, 1NN ED e 1NN DTW, respectivamente. Deve-se notar que para esta análise foram utilizados 23 dos 27 conjuntos de dados avaliados neste capítulo. Os seguintes conjuntos não foram considerados: *CBF*, *Trace*, *SyntheticControl* e *TwoPatterns*, pois não foram avaliados no trabalho do qual os dados

Tabela 19 – Sumário dos resultados da análise *Texas Sharpshooter Fallacy*.

	VP	VN	FP	FN
TS+CFS	14	2	5	1
TS+CS	14	2	2	2
TS+FCBF	8	1	10	4
1NN ED	16	0	6	0
1NN DTW	16	2	4	0

foram coletados.

Os resultados da análise *Texas Sharpshooter Fallacy* apresentados, evidenciam a capacidade do algoritmo HID, em possibilitar a estimativa correta de desempenho para a maioria dos conjuntos de dados avaliados, quando comparado aos demais algoritmos analisados. A exceção refere-se ao algoritmo TS+FCBF, para o qual as estimativas de desempenho foram corretas somente em 39% dos conjuntos.

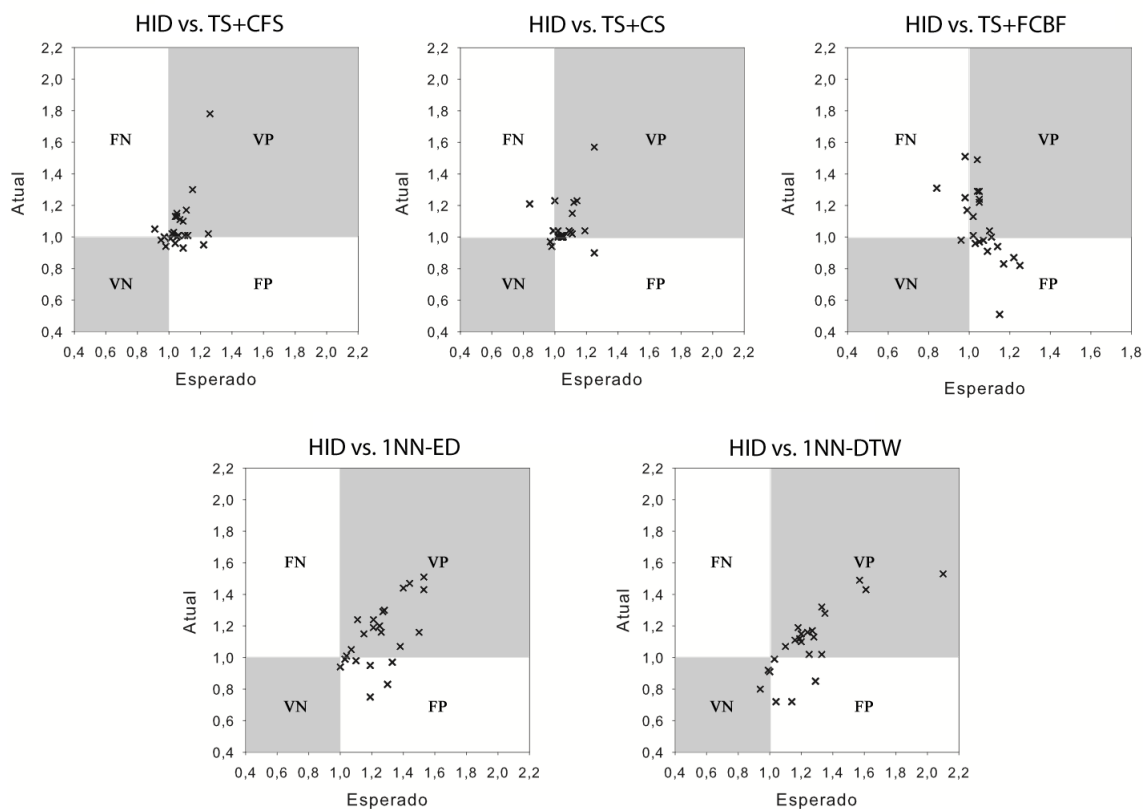


Figura 44 – Comparação entre o algoritmo HID e os algoritmos da abordagem embutida (LS e FS), da abordagem transformada (TS+CS, TS+CFS e TS+FCBF) e do estado da arte (1NN ED e 1NN DTW), pela análise *Texas Sharpshooter Fallacy*.

Com base nos resultados e análises apresentadas nesta avaliação, é possível observar que para os 27 conjuntos de dados utilizados na análise estatística, o método proposto por meio do algoritmo HID apresentou melhor desempenho em relação às demais propostas de construção de modelos simbólicos. De modo que, em relação aos algoritmos baseados na primitiva *shapelet* pela abordagem embutida, HID demonstrou

superioridade estatística, em especial sobre o algoritmo *fast shapelets* (FS), o qual é tido como o estado da arte na construção de modelos simbólicos para a classificação de séries temporais. Em relação aos algoritmos baseados na abordagem da transformada *shapelet*, não foram verificadas diferenças significativas, no entanto, HID apresentou melhor desempenho em termos de *rank* médio e acurácia média. Dado esse cenário, ressalta-se o fato de que o algoritmo HID utiliza somente atributos discretos (0 ou 1) para a indução dos modelos de classificação, ao contrário dos demais algoritmos avaliados, que baseiam-se em uma representação de valores reais da transformada *shapelet*.

Em outro aspecto, é importante constar que, apesar do algoritmo HID consistir em uma abordagem híbrida de classificação, quando um novo exemplo a ser classificado não estiver mapeado no conjunto de treinamento, é possível associá-lo a um caminho (regra) existente no diagrama de decisão, que possa ser o mais similar a esse exemplo. E portanto, é possível interpretar o raciocínio empregado pelo modelo mesmo quando a classificação ocorre por uma estratégia *lazy*, como o 1NN.

Considerando os algoritmos utilizados como referenciais do estado da arte para a classificação de séries temporais, apesar de semelhantes 1NN ED e 1NN DTW, o algoritmo HID apresentou diferença estatística significativa somente em relação ao primeiro. No entanto, mesmo na comparação com 1NN DTW, o algoritmo HID apresentou melhor desempenho, em termos de *rank* médio e acurácia média. Adicionalmente, pelo melhor do nosso conhecimento, o algoritmo HID em combinação estratégia de binarização Pr-TS+CFS apresentou os melhores resultados de acurácia da literatura para os conjuntos de dados *Adiac* (89,77%), *MedicalImages* (81,71%) e *OliveOil* (100,00%). Cabe também ressaltar que resultados interessantes do algoritmo HID foram observados por meio de outras estratégias de binarização.

De modo geral, os resultados e as análises desta avaliação demonstram a possibilidade de se obter bons resultados enquanto preserva-se a capacidade de interpretação do raciocínio utilizado pelos algoritmos de classificação.

## 5.7 REPRESENTAÇÃO SIMBÓLICA HID

Nesta seção são ilustrados os modelos construídos por meio da modelagem parcial proposta no algoritmo HID, para três conjuntos de dados utilizados na avaliação experimental deste capítulo. A representação dos modelos híbridos é como a descrita na seção 5.5, na qual as linhas tracejadas e contínuas representam *ausência* (valor falso 0) e *presença* (valor verdadeiro 1), respectivamente, de uma determinada *shapelet*. Os vértices descritos como círculos representam decisões sobre alguma *shapelet*, e os vértices descritos por retângulos designam as classes. A exceção é o vértice 1NN, que corresponde ao caso quando um possível exemplo não está mapeado pela

estrutura do modelo. Nessa situação a classificação é realizada de modo *lazy*, por meio do algoritmo 1NN (distância Euclidiana). Nas figuras a seguir são representadas as árvores de decisão híbridas e as *shapelets* utilizadas.

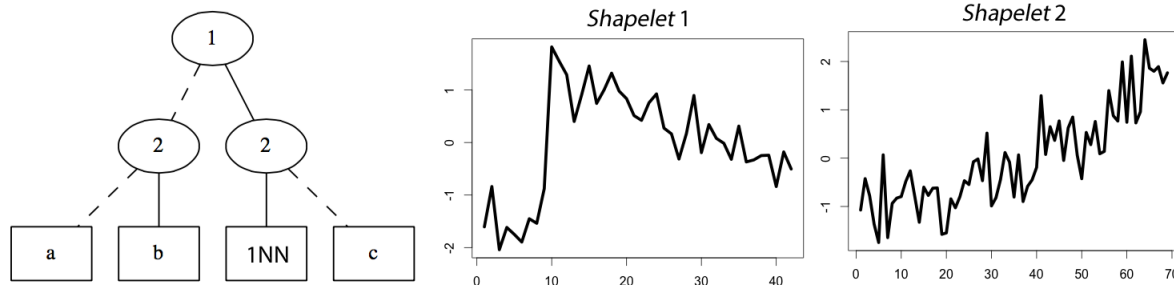


Figura 45 – Representação da árvore de decisão gerada pelo algoritmo HID para o conjunto dados *CBF*.

Na figura 45 é apresentada a árvore de decisão para conjunto de dados *CBF* (#3), que é composto por 30 exemplos no conjunto de treinamento, distribuídos em três classes  $\{a, b, c\}$ , nas proporções 33,33%, 40,00% e 26,67%, respectivamente. Para a árvore de decisão construída pelo algoritmo HID foram utilizadas duas *shapelets*  $\{1, 2\}$ . Como pode ser observado, nessa modelagem há somente uma situação na qual o algoritmo HID utiliza o algoritmo 1NN.

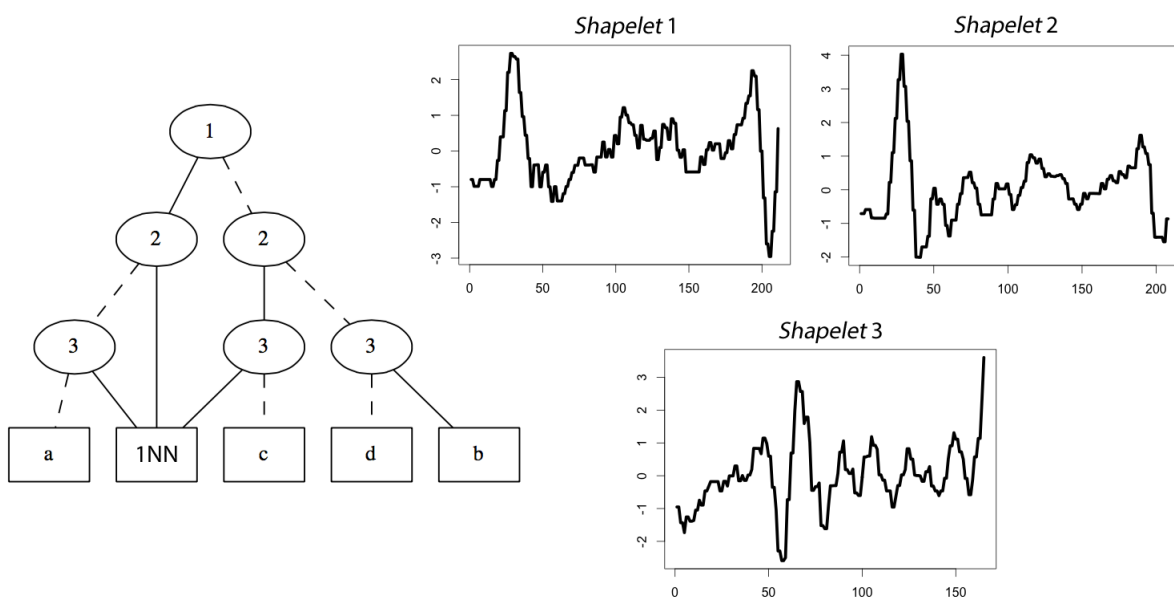


Figura 46 – Representação da árvore de decisão gerada para o conjunto dados *FaceFour*.

Na figura 46 é apresentada a árvore de decisão para o problema *FaceFour* (#11), que é composto por 24 exemplos no conjunto de treinamento, distribuídos em quatro classes  $\{a, b, c, d\}$ , nas proporções 33,33%, 33,33%, 12,50% e 20,83%, respectivamente. Para a árvore de decisão construída pelo algoritmo HID foram utilizadas

três *shapelets* {1, 2, 3}. Nessa modelagem, existem três possíveis exemplos que devem ser classificados pelo algoritmo 1NN.

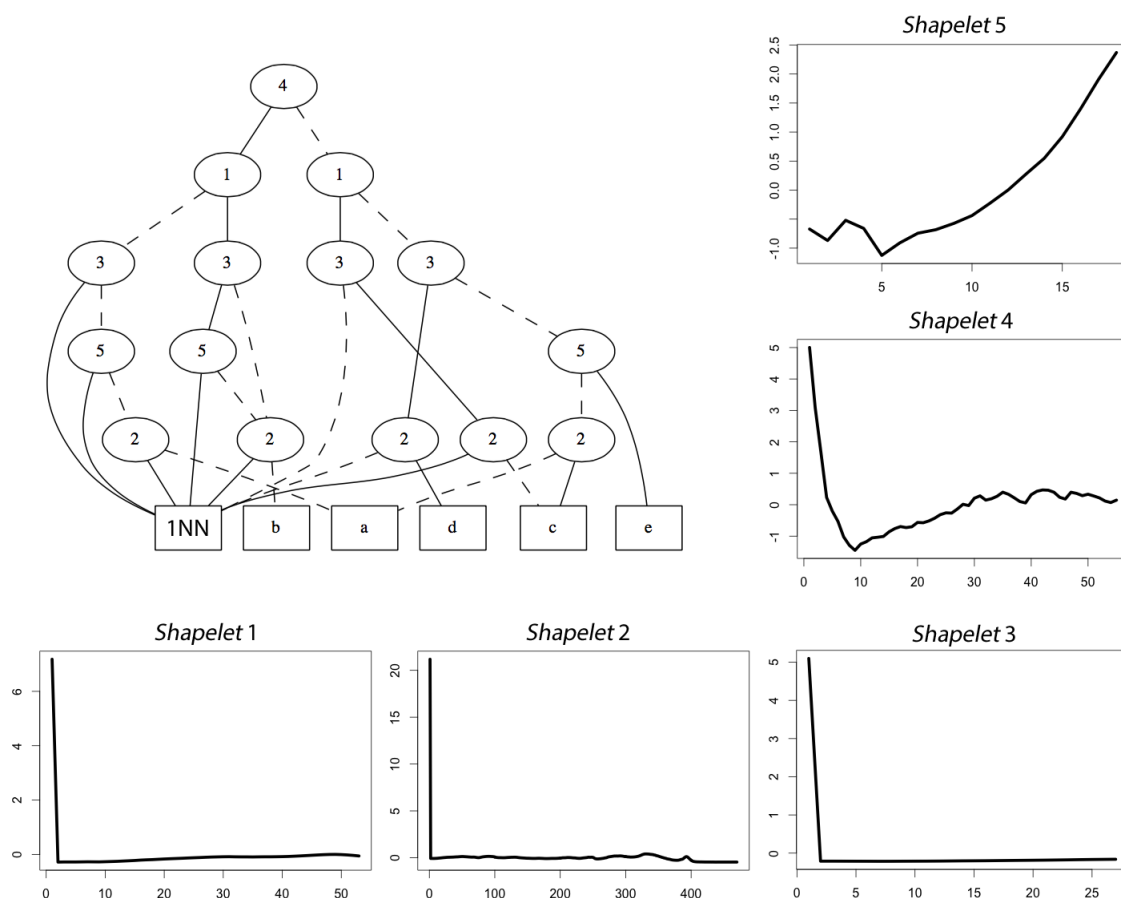


Figura 47 – Representação da árvore de decisão gerada para o conjunto dados *Beef*.

Na figura 47 é apresentada a árvore de decisão para o problema *Beef* (#2), que é composto por 30 exemplos no conjunto de treinamento, distribuídos em cinco classes {a, b, c, d, e}, com 6 exemplos para cada uma. Para a árvore de decisão construída pelo algoritmo HID foram utilizadas cinco *shapelets* {1, 2, 3, 4, 5}. Nesse modelo, existem oito possíveis casos não mapeados pela estrutura da árvore, e que portanto devem ser classificados por meio do algoritmo 1NN.

## 5.8 CONSIDERAÇÕES FINAIS

Neste capítulo foi proposto um método para a construção de modelos simbólicos de classificação de séries temporais, por meio da descrição de *eventos* que representam características morfológicas. Para implementar esse método, foram propostos os algoritmos *binary shapelet transform* (BST) e *hybrid instance based decision tree* (HID). O algoritmo BST foi proposto para converter os valores reais da transformada *shapelet* em valores binários. Esse algoritmo, apesar de simplificar as informações da transformada *shapelet*, possibilita um melhor nível de inteligibilidade, devido à natureza

binária (*presença* ou *ausência*) da representação; e a eliminação de possíveis inconsistências conceituais, que podem se originar quando uma *shapelet* ocorre em mais de uma regra com distintos limiares de separação. Além disso, foram propostas três estratégias para a etapa de aprendizado de limiares do algoritmo BST: precisão (Pr), ganho de informação (Ig) e agrupamento (2k).

O algoritmo HID consiste em uma abordagem híbrida que foi proposta para combinar a característica de inteligibilidade dos modelos baseados em regras, como árvores de decisão, com algoritmos baseados em exemplos, como o algoritmo 1NN. O algoritmo HID utiliza diagramas algébricos de decisão por meio da representação binária de *eventos* em séries temporais. Neste trabalho, foi proposta a utilização de descritores morfológicos (*shapelets*) para a caracterização de *eventos*. Como descrito nos capítulos anteriores, a primitiva *shapelet* tem possibilitado o desenvolvimento de métodos de melhor desempenho, em termos de acurácia e de inteligibilidade, para a classificação de séries temporais.

Com o intuito de verificar o desempenho do método proposto, foi conduzida uma avaliação experimental utilizando conjuntos de dados amplamente estudados na literatura de séries temporais. A análise experimental envolveu algoritmos baseados na construção de modelos simbólicos e híbridos, bem como métodos considerados estado da arte para a classificação de séries temporais, mas que não produzem modelos sobre o conjunto de treinamento. Por meio dos resultados experimentais sobre o conjunto de dados utilizado, verificou-se o desempenho superior do algoritmo HID em relação ao J48, e sua competitividade em relação aos modelos híbridos produzidos pelo algoritmo NNGE. Em comparação com as abordagens estado da arte, LS e FS, para a construção modelos simbólicos, o algoritmo HID apresentou desempenho superior em relação a ambas. Na análise dos algoritmos não simbólicos verificou-se superior desempenho do algoritmo HID em relação ao 1NN ED. Em relação ao 1NN DTW, apesar de resultados melhores do algoritmo HID, não foi verificada diferença estatística.

Sob uma perspectiva geral dos resultados e avaliações realizadas neste capítulo, pode-se verificar que existe a possibilidade de construção de modelos inteligíveis que tenham desempenho equivalente ou até mesmo superior aos algoritmos que não produzem modelos simbólicos.

No próximo capítulo, são apresentadas as principais contribuições deste trabalho, bem como as limitações identificadas e os trabalhos futuros.

## 6 CONCLUSÃO

Neste capítulo são apresentadas as conclusões decorrentes desta tese de doutorado. Os objetivos do trabalho são retomados e relacionados com os principais resultados e contribuições correspondentes. Posteriormente são destacadas as limitações e os trabalhos futuros.

O objetivo geral deste trabalho consistiu na exploração da tarefa de classificação de séries temporais, considerando a utilização de descritores de características morfológicas para a construção de modelos simbólicos. Especificamente, dentre os problemas estudados, destacam-se as seguintes contribuições:

**Capítulo 4:** melhoria da qualidade de representação da transformada *shapelet*, considerando a construção de modelos de classificação simbólicos baseados em árvores de decisão.

**Capítulo 5:** construção de modelos simbólicos de classificação por meio de uma abordagem híbrida, combinando a representação de árvore de decisão com o algoritmo vizinho mais próximo.

### 6.1 EXPLORANDO A TRANSFORMA SHAPELET - CAPÍTULO 4

No capítulo 4 foi explorada a utilização da transformada *shapelet* (LINES et al., 2012) para a representação de séries temporais em problemas de classificação, considerando a construção de modelos por meio de árvores de decisão. Para isso, foram propostas três abordagens para contornar algumas das desvantagens do algoritmo original da transformada *shapelet* (TS): exaustiva (TSk), relaxada (TS\*) e reduzida (TS+). Com o intuito de verificar o desempenho dessas propostas, foi conduzida uma avaliação experimental, na qual cada uma delas foi comparada com TS. Baseando-se nos resultados experimentais dessa avaliação, foi possível realizar as seguintes observações:

- Embora não tenha sido constatada diferença estatística, TSk apresentou melhor valor de *rank* médio em relação a TS. A partir dessa análise verifica-se a possibilidade de que *shapelets* de melhor representatividade não sejam selecionadas quando se limita de modo inferior ou superior o tamanho das subsequências a serem avaliadas;
- Na comparação de TS\* com TS não foi verificada diferença estatística, no entanto, em termos de *rank* médio TS\* apresentou melhor desempenho. Esse resultado



evidência que a estimativa inadequada do parâmetro  $k$  pode prejudicar a identificação de *shapelets* que possam melhor representar cada uma das classes do domínio avaliado. Contudo, a utilização de todos os tamanhos de subsequência possíveis, como proposto na abordagem TS\*, pode causar um sobreajuste nos modelos induzidos, em particular com árvores de decisão (J48). Essa constatação pôde ser evidenciada pela comparação de TS\* e TSk, na qual TSk apresentou melhor desempenho em termos de *rank* médio. Essa constatação corrobora com as ponderações sobre a TS, apresentadas em [Hills et al. \(2013\)](#);

- Na comparação de TS+ com TS, as variações TS+CS e TS+FCBF apresentaram resultados estatisticamente superiores em relação à TS, e embora para a variação TS+CBF não tenha sido verificada diferença significativa, esta apresentou melhor valor de *rank* médio em comparação à TS. Com base nesses resultados pode-se concluir que a abordagem TS+ consiste em uma alternativa mais adequada para a determinação automática de parâmetros, como tamanho mínimo e máximo de subsequência e a quantidade de *shapelets* a serem utilizadas, e também como estratégia para evitar o sobreajuste da abordagem TS\*.

A partir da análise dos resultados referentes às três abordagens propostas, ainda no capítulo 4 foi conduzida uma avaliação experimental para analisar o desempenho da abordagem TS+ em relação aos algoritmos da abordagem embutida, *logical shapelets* e *fast shapelets*, e aos algoritmos estado da arte, 1NN ED e 1NN DTW. Nessa análise, não foi verificada diferença estatística de TS+ em relação às demais abordagens. No entanto, as variações de TS+ apresentaram valores de *rank* médio superiores a todos os demais algoritmos avaliados, com exceção do 1NN DTW. Esses resultados evidenciam a capacidade promissora de TS+ para a classificação de séries temporais.

Com base nos resultados experimentais apresentados nesse capítulo, pode-se concluir que a qualidade da representação da transformada *shapelet* tem influência no desempenho de classificadores simbólicos como árvores de decisão. Portanto, o desenvolvimento de técnicas para a melhoria do processo de identificação de *shapelets* pode contribuir para a construção de classificadores inteligíveis e competitivos em relação aos métodos não simbólicos, em termos de acurácia.

## 6.2 ABORDAGEM HÍBRIDA EM DIAGRAMAS DE DECISÃO - CAPÍTULO 5

No capítulo 5 foi proposto um método para a construção de modelos simbólicos de classificação de séries temporais, por meio da descrição de *eventos* que representam características morfológicas. Para implementar esse método, neste trabalho foram propostos os algoritmos *binary shapelet transform* (BST) e *hybrid instance based deci-*

*sion tree* (HID), e considerando as contribuições discutidas no capítulo 4, foi proposto utilizar a transformada *shapelet* para a descrição de *eventos*.

O algoritmo BST foi proposto para converter os valores contínuos da transformada *shapelet* em valores binários. Desse modo, cada série temporal passa a ser representada por um conjunto de valores que indicam a *presença* ou a *ausência* das *shapelets* selecionadas. Apesar de simplificar as informações da transformada *shapelet*, algumas vantagens da utilização da representação binária provida pelo BST podem ser ressaltadas:

- Possibilidade de prover melhor nível de inteligibilidade, devido a natureza binária (*presença* ou *ausência*) da representação; e
- A capacidade de eliminar possíveis inconsistências conceituais, que podem se originar quando uma *shapelet* ocorre em mais de uma regra com distintos limiares de separação.

Motivado pela ideia de combinar a capacidade de inteligibilidade dos modelos baseados em regras com algoritmos baseados em exemplos (SALZBERG, 1991; FRIEDMAN, 1996), foi proposto o algoritmo HID. Este algoritmo consiste em uma abordagem híbrida que utiliza a representação binária do BST para a construção de modelos simbólicos por meio de diagramas algébricos de decisão (ADD) (BAHAR et al., 1993) juntamente com o algoritmo 1NN.

Com o intuito de verificar o desempenho do método proposto, foi conduzida uma avaliação experimental do algoritmo HID, utilizando conjuntos de dados amplamente estudados na literatura de séries temporais. Baseando-se nos resultados experimentais dessa avaliação, foi possível verificar as seguintes observações:

- O algoritmo HID apresentou desempenho estatisticamente superior em relação ao J48. Em relação à comparação de HID com os modelos híbridos produzidos pelo algoritmo NNGE, não foi verificada diferença estatística, no entanto, os valores de *rank* médio de HID foram superiores ao de NNGE. Com base nesses resultados pode-se concluir que o método proposto, por meio do algoritmo HID, consiste na alternativa mais adequada, dentre os algoritmos avaliados, para a construção de modelos simbólicos utilizando a representação binária provida pelo BST;
- Na comparação com os algoritmos da transformada *shapelet* reduzida (TS+), propostos no capítulo 4, o algoritmo HID não foi estatisticamente superior, no entanto HID apresentou os melhores resultados em termos de *rank* médio. Com base nessa análise, verifica-se que embora o algoritmo HID utilize como base

uma representação binária, e portanto mais simples, os resultados obtidos demonstram sua competitividade em relação às abordagens que utilizam os valores de distância;

- Em comparação com as abordagens estado da arte para a construção de modelos simbólicos (LS e FS), o algoritmo HID apresentou desempenho estatisticamente superior em relação a ambas. Baseando-se nesses resultados, verifica-se que HID pode ser a melhor alternativa para a construção de modelos simbólicos de classificação de séries temporais;
- Na análise dos algoritmos não simbólicos, verificou-se um superior desempenho do algoritmo HID em relação ao 1NN ED. Em relação ao 1NN DTW, apesar de resultados melhores do algoritmo HID, não foi verificada diferença estatística. Com base nessa análise pode-se concluir que métodos híbridos podem contribuir para a construção de modelos simbólicos e apresentar desempenho equivalente ou até mesmo superior aos métodos não simbólicos.

Sob uma perspectiva geral dos resultados e avaliações realizadas nesta tese, é possível sustentar que existe um caminho promissor para o desenvolvimento de métodos que permitam a construção de modelos inteligíveis para a classificação de séries temporais, e que apresentem desempenho equivalente ou até mesmo superior aos algoritmos que não produzem modelos simbólicos.

### 6.3 LIMITAÇÕES

As limitações identificadas em relação aos métodos e algoritmos propostos neste trabalho referem-se aos seguintes aspectos:

- O custo computacional de tempo para a construção de um ADD mínimo é exponencial em função da quantidade de *shapelets* utilizadas. No entanto, espera-se que para a construção de modelos que possam ser inteligíveis, a quantidade de *shapelets* seja a menor possível. Adicionalmente, existem heurísticas para a construção de ADDs que permitem reduzir esse custo para grande parte dos domínios;
- Determinados tipos de comportamentos podem não ser adequadamente representados por meio padrões morfológicos. Alguns padrões, por exemplo, são melhores caracterizados quando mapeados no domínio de frequências;
- O custo computacional do algoritmo de identificação de *shapelets* pode se tornar inviável em determinados domínios. Outras abordagens mais eficientes, em

termos de tempo, podem ser aplicadas, no entanto a maioria delas inserem a necessidade de estimativa de parâmetros ou algoritmos não determinísticos;

- A abordagem atual do método proposto, em particular relacionado ao algoritmo HID, não possibilita descrição de relacionamentos de ordem ou de tempo entre as *shapelets*. Esse tipo de informação pode ser de fundamental importância em diversos domínios de séries temporais;
- O algoritmo HID considera o mesmo grau de importância para todas as *shapelets* da representação binária, o que pode diluir a importância de possíveis *shapelets* mais significativas, e portanto prejudicar o desempenho preditivo do algoritmo.

## 6.4 TRABALHOS FUTUROS

Durante o desenvolvimento deste trabalho foram identificados diversos aspectos a serem explorados em estudos futuros, tais como:

- Adaptação da abordagem proposta para aplicação no problema de classificação precoce (*early classification*) (XING et al., 2011), o qual tem sido recentemente explorado na literatura por meio do uso da primitiva *shapelet*;
- Estudo e desenvolvimento de novos algoritmos para a etapa de aprendizado de limiares do algoritmo BST, e a adaptação de outros algoritmos de aprendizado para a função de aproximação do algoritmo HID;
- Explorar, por meio do método proposto, outras abordagens para a caracterização de eventos usando padrões morfológicos. Uma possibilidade seria a utilização de estratégias não supervisionadas, tais como a identificação de *motifs*;
- Adaptação do método proposto para o tratamento de possíveis relações de ordem ou de tempo existentes entre as *shapelets*. Uma possibilidade seria a utilização de algoritmos como regras de associação sequencial (AGRAWAL; SRIKANT, 1995);
- Avaliar as variações propostas para a transformada *shapelet* por meio de outras medidas de qualidade (LINES et al., 2012), bem como com outras medidas de similaridade e outros algoritmos tradicionais de aprendizado simbólico;
- Baseando-se na estratégia adotada pelo algoritmo NNGE, uma possibilidade de ressaltar padrões de maior importância seria por meio da atribuição de pesos para cada *shapelet*;
- Devido ao fato de que muitos dos conjuntos de dados analisados apresentam desbalanceamento de classes entre os exemplos, a utilização de outras medidas

de avaliação de modelos, tais como medida-f, poderá fornecer melhor estimativa de desempenho do que somente a medida de acurácia;

- Utilizar métricas de complexidade de regras para avaliar os modelos produzidos pelos diferentes algoritmos de aprendizado simbólico analisados;
- Analisar os domínios dos conjuntos de dados avaliados, para os quais cada método de binarização e de seleção de atributos apresentaram melhor e pior desempenho.

## REFERÊNCIAS

- AGRAWAL, R.; FALOUTSOS, C.; SWAMI, A. N. Efficient similarity search in sequence databases. In: **Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms**. London, UK, UK: Springer-Verlag, 1993. (FODO '93), p. 69–84. ISBN 3-540-57301-1. Citado 2 vezes nas páginas 41 e 47.
- AGRAWAL, R.; IMIELIŃSKI, T.; SWAMI, A. Mining association rules between sets of items in large databases. In: **Proceedings of the 1993 ACM SIGMOD international conference on Management of data**. New York, NY, USA: ACM, 1993. (SIGMOD '93), p. 207–216. ISBN 0-89791-592-5. Citado na página 49.
- AGRAWAL, R. et al. Querying shapes of histories. In: **Proceedings of the 21th International Conference on Very Large Data Bases**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995. (VLDB '95), p. 502–514. ISBN 1-55860-379-4. Citado na página 43.
- AGRAWAL, R.; SRIKANT, R. Fast algorithms for mining association rules in large databases. In: **Proceedings of the 20th International Conference on Very Large Data Bases**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994. (VLDB '94), p. 487–499. ISBN 1-55860-153-8. Citado na página 49.
- AGRAWAL, R.; SRIKANT, R. Mining sequential patterns. In: **Proceedings of the Eleventh International Conference on Data Engineering**. Washington, DC, USA: IEEE Computer Society, 1995. (ICDE '95), p. 3–14. ISBN 0-8186-6910-1. Citado 2 vezes nas páginas 42 e 147.
- AHA, D. W.; KIBLER, D.; ALBERT, M. K. Instance-based learning algorithms. **Machine Learning** 6, v. 6, p. 37–66, 1991. Citado na página 34.
- AHMED, N. K. et al. An empirical comparison of machine learning models for time series forecasting. **Econometric Reviews**, v. 29, n. 5-6, p. 594–621, 2010. Citado na página 50.
- AKERS, S. Binary decision diagrams. **Computers, IEEE Transactions on**, C-27, n. 6, p. 509–516, June 1978. ISSN 0018-9340. Citado na página 111.
- ALONSO, F. et al. Modelling medical time series using grammar-guided genetic programming. In: **ICDM**. [S.l.: s.n.], 2008. p. 32–46. Citado na página 42.
- ALPAYDIN, E. **Introduction to Machine Learning**. Cambridge, England: MIT Press, 2004. Citado na página 32.
- ANDERSEN, H. R. **An introduction to binary decision diagrams**. [S.l.], 1997. Citado na página 112.
- ANTUNES, C. M.; OLIVEIRA, A. L. Temporal data mining: an overview. In: **KDD Workshop on Temporal Data Mining**. [S.l.: s.n.], 2001. p. 1–13. Citado 2 vezes nas páginas 25 e 37.

ASTROM, K. J. Information science: On the choice of sampling rates in parametric identification of time series. **Inf. Sci.**, Elsevier Science Inc., New York, NY, USA, v. 1, n. 3, p. 273–278, jul. 1969. ISSN 0020-0255. Citado na página 41.

BAGNALL, A. et al. Transformation based ensembles for time series classification. In: **Proceedings of the Twelfth SIAM International Conference on Data Mining, Anaheim, California, USA, April 26-28, 2012**. [S.l.: s.n.], 2012. p. 307–318. Citado 4 vezes nas páginas 25, 28, 51 e 79.

BAGNALL, A. et al. A bit level representation for time series data mining with shape based similarity. **Data Min. Knowl. Discov.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 13, n. 1, p. 11–40, jul. 2006. ISSN 1384-5810. Citado na página 43.

BAHAR, R. et al. Algebraic decision diagrams and their applications. In: **Computer-Aided Design, 1993. ICCAD-93. Digest of Technical Papers., 1993 IEEE/ACM International Conference on**. [S.l.: s.n.], 1993. p. 188–191. Citado 2 vezes nas páginas 112 e 145.

BAKSHI, B. R. et al. Analysis of operating data for evaluation, diagnosis and control of batch operations. **Journal of Process Control**, v. 4, n. 4, p. 179–194, 1994. ISSN 0959-1524. Citado 5 vezes nas páginas 26, 51, 54, 55 e 103.

BAO, D. A generalized model for financial time series representation and prediction. **Applied Intelligence**, Kluwer Academic Publishers, Hingham, MA, USA, v. 29, n. 1, p. 1–11, ago. 2008. ISSN 0924-669X. Citado na página 42.

BARROS, S. do Lago Pereira e Leliane Nunes de. **Diagramas de Decisão Binária**. [S.l.], 2007. Citado 2 vezes nas páginas 110 e 112.

BATISTA, G. E. A. P. A. et al. Cid: an efficient complexity-invariant distance for time series. **Data Mining and Knowledge Discovery**, Springer US, p. 1–36, 2013. Citado 3 vezes nas páginas 53, 55 e 84.

BATISTA, G. E. A. P. A.; WANG, X.; KEOGH, E. J. A complexity-invariant distance measure for time series. In: **SDM**. [S.l.: s.n.], 2011. p. 699–710. Citado 5 vezes nas páginas 25, 45, 98, 99 e 103.

BATYRSHIN, I. Z.; SHEREMETOV, L. Perception-based approach to time series data mining. **Applied Soft Computing**, Elsevier, v. 8, n. 3, p. 1211–1221, 2008. Citado 2 vezes nas páginas 28 e 55.

BAYDOGAN, M.; RUNGER, G.; TUV, E. A bag-of-features framework to classify time series. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 35, n. 11, p. 2796–2802, Nov 2013. ISSN 0162-8828. Citado na página 52.

BERNDT, D. J.; CLIFFORD, J. Using Dynamic Time Warping to Find Patterns in Time Series. In: **KDD Workshop**. [S.l.: s.n.], 1994. p. 359–370. Citado na página 45.

BOLLIG, B.; WEGENER, I. Improving the variable ordering of obdds is np-complete. **Computers, IEEE Transactions on**, v. 45, n. 9, p. 993–1002, Sep 1996. ISSN 0018-9340. Citado na página 111.



BOX, G. E. P.; JENKINS, C. M.; REINSEL, G. C. **Time Series Analysis: Forecasting and Control**. 3. ed. Englewood – NJ, USA: Prentice hall, 1994. Citado 2 vezes nas páginas 37 e 50.

BREIMAN, L. et al. **Classification and Regression Trees**. [S.l.]: Wadsworth, 1984. ISBN 0-534-98053-8. Citado na página 103.

BROCKWELL, P. J.; DAVIS, R. A. **Introducing to Time Series and Forecasting**. Second. New York, NY-USA: Springer, 1996. Citado na página 35.

BRYANT, R. Graph-based algorithms for boolean function manipulation. **Computers, IEEE Transactions on**, C-35, n. 8, p. 677–691, Aug 1986. ISSN 0018-9340. Citado na página 111.

BUHLER, J.; TOMPA, M. Finding motifs using random projections. **Journal of Computational Biology**, v. 9, n. 2, p. 225–242, 2002. Citado 3 vezes nas páginas 69, 71 e 81.

BUZA, K.; SCHMIDT-THIEME, L. Motif-based classification of time series with bayesian networks and svms. In: FINK, A. et al. (Ed.). **Advances in Data Analysis, Data Handling and Business Intelligence**. [S.l.]: Springer Berlin Heidelberg, 2010, (Studies in Classification, Data Analysis, and Knowledge Organization). p. 105–114. ISBN 978-3-642-01043-9. Citado na página 52.

CAMPANA, B. J. L.; KEOGH, E. J. A compression based distance measure for texture. **Stat. Anal. Data Min.**, John Wiley & Sons, Inc., New York, NY, USA, v. 3, n. 6, p. 381–398, dez. 2010. ISSN 1932-1864. Citado na página 53.

CASTRO, N. C. **Time series motif discovery**. Tese (Tese de doutorado) — Universidade do Minho, Minho, Portugal, 2012. Citado 5 vezes nas páginas 37, 38, 40, 45 e 51.

CATLETT, J. On changing continuous attributes into ordered discrete attributes. In: **Proceedings of the European working session on learning on Machine learning**. New York, NY, USA: Springer-Verlag New York, Inc., 1991. (EWSL-91), p. 164–178. ISBN 0-387-53816-X. Citado na página 56.

CHAKRABARTI, K. et al. Locally adaptive dimensionality reduction for indexing large time series databases. **ACM Trans. Database Syst.**, ACM, New York, NY, USA, v. 27, n. 2, p. 188–228, jun. 2002. ISSN 0362-5915. Citado na página 59.

CHAN, F. K.-P.; FU, A. Wai-chee; YU, C. Haar wavelets for efficient similarity search of time-series: With and without time warping. **IEEE Trans. on Knowl. and Data Eng.**, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 15, n. 3, p. 686–705, mar. 2003. ISSN 1041-4347. Citado na página 41.

CHAN, K.-P.; FU, A.-C. Efficient time series matching by wavelets. In: **Data Engineering, 1999. Proceedings., 15th International Conference on**. [S.l.: s.n.], 1999. p. 126–133. ISSN 1063-6382. Citado 2 vezes nas páginas 41 e 47.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 41, n. 3, p. 15:1–15:58, jul. 2009. ISSN 0360-0300. Citado na página 50.



CHANG, K.-W. et al. Efficient pattern-based time series classification on gpu. In: **Proceedings of the 12th IEEE International Conference on Data Mining**. [S.l.: s.n.], 2012. Citado na página 77.

CHEN, J. R. Making subsequence time series clustering meaningful. In: **Proceedings of the Fifth IEEE International Conference on Data Mining**. Washington, DC, USA: IEEE Computer Society, 2005. (ICDM '05), p. 114–121. ISBN 0-7695-2278-5. Citado na página 48.

CHEN, L.; NG, R. On the marriage of lp-norms and edit distance. In: **Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30**. [S.l.]: VLDB Endowment, 2004. (VLDB '04), p. 792–803. ISBN 0-12-088469-0. Citado na página 48.

CHIU, B.; KEOGH, E.; LONARDI, S. Probabilistic discovery of time series motifs. In: **Proceedings of the 9th International Conference on Knowledge Discovery and Data Mining**. New York, USA: ACM Press, 2003. p. 493–498. ISBN 1-58113-737-0. Citado 3 vezes nas páginas 51, 57 e 69.

CHU, S. et al. Iterative deepening dynamic time warping for time series. In: **Second SIAM International Conference on Data Mining**. Arlington, Virginia, USA: [s.n.], 2002. p. 1–18. Citado na página 46.

CORDUAS, M.; PICCOLO, D. Time series clustering and classification by the autoregressive metric. **Computational Statistics Data Analysis**, v. 52, n. 4, p. 1860 – 1872, 2008. ISSN 0167-9473. Citado na página 49.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, n. 3, p. 273–297, 1995. ISSN 0885-6125. Citado na página 34.

COTOFREI, P.; STOFFEL, K. Rule extraction from time series databases using classification trees. In: **Proceedings of the 20th International Association of Science and Technology for Development Conference on Applied Informatics**. Innsbruck, Austria: ACTA Press, 2002. p. 327–332. Citado 5 vezes nas páginas 25, 26, 54, 56 e 103.

COTOFREI, P.; STOFFEL, K. First-order logic based formalism for temporal data mining \*. In: **Foundations of Data Mining and knowledge Discovery**. [S.l.: s.n.], 2005. p. 185–210. Citado 4 vezes nas páginas 26, 54, 56 e 103.

DAS, G.; GUNOPULOS, D.; MANNILA, H. Finding similar time series. In: **Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery**. London, UK, UK: Springer-Verlag, 1997. (PKDD '97), p. 88–100. ISBN 3-540-63223-9. Citado na página 48.

DAS, G. et al. Rule discovery from time series. **Knowledge Discovery and Data Mining**, p. 16–22, 1998. Citado 2 vezes nas páginas 43 e 50.

DAW, C. S.; FINNEY, C. E. A.; TRACY, E. R. A review of symbolic analysis of experimental data. **Review of Scientific Instruments**, v. 74, n. 2, p. 915–930, 2003. Citado na página 42.

DELGADO, K. V. **Processos de decisão Markovianos fatorados com probabilidades imprecisas**. Tese (Tese de doutorado) — Instituto de Matemática e Estatística–USP, São Carlos, São Paulo, Brasil, 2010. Citado 4 vezes nas páginas 111, 112, 113 e 114.

DEMSAR, J. Statistical comparisons of classifiers over multiple data sets. **Journal of Machine Learning Research**, v. 7, p. 1–30, 2006. Citado 4 vezes nas páginas 87, 88, 90 e 123.

DENG, H. et al. A time series forest for classification and feature extraction. **Information Sciences**, v. 239, n. 0, p. 142 – 153, 2013. ISSN 0020-0255. Citado na página 52.

DENG, K.; MOORE, A. W.; NECHYBA, M. C. Learning to recognize time series: Combining arma models with memory-based learning. In: **Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation**. Washington, DC, USA: IEEE Computer Society, 1997. (CIRA '97), p. 246–250. ISBN 0-8186-8138-1. Citado na página 52.

DING, H. et al. Querying and mining of time series data: experimental comparison of representations and distance measures. **PVLDB**, v. 1, n. 2, p. 1542–1552, 2008. Citado 5 vezes nas páginas 25, 53, 55, 103 e 122.

DOMINGOS, P. Rule induction and instance-based learning a unified approach. In: **Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995. (IJCAI'95), p. 1226–1232. ISBN 1-55860-363-8. Citado na página 27.

DRECHSLER, R.; BECKER, B.; GOCKEL, N. Genetic algorithm for variable ordering of obdds. **Computers and Digital Techniques, IEE Proceedings -**, v. 143, n. 6, p. 364–368, Nov 1996. ISSN 1350-2387. Citado na página 111.

DUNN, J.; DUNN, O. J. Multiple comparisons among means. **American Statistical Association**, p. 52–64, 1961. Citado na página 88.

EHLERS, R. S. **Análise de Séries Temporais**. Curitiba — PR, Brasil, 2005. Citado 5 vezes nas páginas 25, 31, 34, 36 e 37.

ESLING, P.; AGON, C. Time-series data mining. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 45, n. 1, p. 12:1–12:34, dez. 2012. ISSN 0360-0300. Citado 9 vezes nas páginas 25, 31, 37, 40, 44, 47, 48, 49 e 50.

FACELI, R. et al. **Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina**. Rio de Janeiro, Brasil: LTC, 2011. Citado 7 vezes nas páginas 26, 27, 28, 32, 33, 49 e 55.

FALOUTSOS, C.; RANGANATHAN, M.; MANOLOPOULOS, Y. Fast subsequence matching in time-series databases. In: **Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: ACM, 1994. (SIGMOD '94), p. 419–429. ISBN 0-89791-639-5. Citado na página 40.

FAYYAD, U.; PIATETSKY-SHAPIO, G.; SMYTH, P. Knowledge discovery and data mining: Towards a unifying framework. In: **Proceedings of the Second International Conference on Knowledge Discovery and Data Mining**. Menlo Park, USA: [s.n.], 1996. p. 82–88. Citado 2 vezes nas páginas 31 e 32.

FERN, X. Z.; BRODLEY, C. E. Boosting lazy decision trees. In: **Proceedings of the Twentieth International Conference on Machine Learning, August 2003 Washington D.C.** [S.l.: s.n.], 2003. p. 178–185. Citado 3 vezes nas páginas 27, 29 e 104.

FERRERO, C. A. **Algoritmo kNN para previsão de dados temporais: funções de previsão e critérios de seleção de vizinhos próximos aplicados a variáveis ambientais em limnologia.** Dissertação (Mestrado) — Universidade de São Paulo - USP, 2009. Citado 5 vezes nas páginas 25, 31, 35, 36 e 50.

FRIEDMAN, J. H. Lazy decision trees. In: **Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1.** [S.l.]: AAAI Press, 1996. (AAAI'96), p. 717–724. ISBN 0-262-51091-X. Citado 5 vezes nas páginas 27, 29, 103, 104 e 145.

FU, A.-C. et al. Scaling and time warping in time series querying. **The VLDB Journal**, Springer-Verlag, v. 17, n. 4, p. 899–921, 2008. ISSN 1066-8888. Citado na página 45.

FU, T. chung. A review on time series data mining. **Eng. Appl. of AI**, v. 24, n. 1, p. 164–181, 2011. Citado 4 vezes nas páginas 25, 31, 37 e 40.

FUJITA, M.; MATSUNAGA, Y.; KAKUDA, T. On variable ordering of binary decision diagrams for the application of multi-level logic synthesis. In: **Design Automation. EDAC., Proceedings of the European Conference on.** [S.l.: s.n.], 1991. p. 50–54. Citado na página 111.

GAFFNEY, S.; SMYTH, P. Trajectory clustering with mixtures of regression models. In: **Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.** New York, NY, USA: ACM, 1999. (KDD '99), p. 63–72. ISBN 1-58113-143-7. Citado na página 48.

GAO, B. J.; ESTER, M. Right of inference: Nearest rectangle learning revisited. In: **Proceedings of the 17th European Conference on Machine Learning.** Berlin, Heidelberg: Springer-Verlag, 2006. (ECML'06), p. 638–645. ISBN 3-540-45375-X, 978-3-540-45375-8. Citado 3 vezes nas páginas 27, 29 e 104.

GEURTS, P. Pattern extraction for time series classification. In: **Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery.** London, UK: Springer-Verlag, 2001. p. 115–127. ISBN 3-540-42534-9. Citado 10 vezes nas páginas 25, 26, 28, 42, 55, 57, 58, 59, 60 e 103.

GHALWASH, M.; RADOSAVLJEVIC, V.; OBRADOVIC, Z. Extraction of interpretable multivariate patterns for early diagnostics. In: **Data Mining (ICDM), 2013 IEEE 13th International Conference on.** [S.l.: s.n.], 2013. p. 201–210. ISSN 1550-4786. Citado 5 vezes nas páginas 26, 53, 55, 76 e 77.

GHALWASH, M. F.; OBRADOVIC, Z. Early classification of multivariate temporal observations by extraction of interpretable shapelets. **BMC Bioinformatics**, BioMed Central, v. 13, n. 1, 2012. Citado 5 vezes nas páginas 26, 53, 55, 66 e 76.

GHALWASH, M. F.; RADOSAVLJEVIC, V.; OBRADOVIC, Z. Utilizing temporal patterns for estimating uncertainty in interpretable early decision making. In: **Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery**

**and Data Mining**. New York, NY, USA: ACM, 2014. (KDD '14), p. 402–411. ISBN 978-1-4503-2956-9. Citado 3 vezes nas páginas 53, 55 e 77.

GILES, C. L.; LAWRENCE, S.; TSOI, A. C. Noisy time series prediction using recurrent neural networks and grammatical inference. **Mach. Learn.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 44, n. 1-2, p. 161–183, jul. 2001. ISSN 0885-6125. Citado na página 43.

GIUSTI, R.; BATISTA, G. E. A. P. A. An empirical comparison of dissimilarity measures for time series classification. In: **Proceedings of the 2013 Brazilian Conference on Intelligent Systems**. Washington, DC, USA: IEEE Computer Society, 2013. (BRACIS '13), p. 82–88. ISBN 978-0-7695-5092-3. Citado na página 53.

GONZALEZ, J. A.; DIEZ, J. J. R. Boosting interval-based literals: Variable length and early classification. **Data mining in time series databases**. World Scientific, 2004. Citado na página 56.

GORDON, D.; HENDLER, D.; ROKACH, L. Fast randomized model generation for shapelet-based time series classification. **CoRR**, abs/1209.5038, 2012. Citado 3 vezes nas páginas 26, 27 e 77.

GRABOCKA, J.; NANOPOULOS, A.; SCHMIDT-THIEME, L. Invariant time-series classification. In: FLACH, P.; BIE, T. D.; CRISTIANINI, N. (Ed.). **Machine Learning and Knowledge Discovery in Databases**. [S.l.]: Springer Berlin Heidelberg, 2012, (Lecture Notes in Computer Science, v. 7524). p. 725–740. ISBN 978-3-642-33485-6. Citado na página 53.

GRABOCKA, J. et al. Learning time-series shapelets. In: **Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2014. (KDD '14), p. 392–401. ISBN 978-1-4503-2956-9. Citado 3 vezes nas páginas 26, 27 e 77.

GRABOCKA, J.; WISTUBA, M.; SCHMIDT-THIEME, L. Scalable Discovery of Time-Series Shapelets. **ArXiv e-prints**, mar. 2015. Citado 3 vezes nas páginas 26, 27 e 77.

GUDMUNDSSON, S.; RUNARSSON, T.; SIGURDSSON, S. Support vector machines and dynamic time warping for time series. In: **Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on**. [S.l.: s.n.], 2008. p. 2772–2776. ISSN 1098-7576. Citado na página 53.

GULLO, F. et al. A time series representation model for accurate and fast similarity detection. **Pattern Recogn.**, Elsevier Science Inc., New York, NY, USA, v. 42, n. 11, p. 2998–3014, nov. 2009. ISSN 0031-3203. Citado na página 42.

HALL, M. et al. The weka data mining software: an update. **ACM SIGKDD Explorations Newsletter**, ACM, v. 11, n. 1, p. 10–18, 2009. Citado na página 84.

HALL, M. A. Correlation-based feature selection for discrete and numeric class machine learning. In: **Proceedings of the Seventeenth International Conference on Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000. (ICML '00), p. 359–366. ISBN 1-55860-707-2. Citado na página 83.

HAN, J.; KAMBER, M. **Data Mining: concepts and techniques**. 2. ed. San Francisco, California, USA: Morgan Kaufmann, 2006. Citado 3 vezes nas páginas 31, 38 e 49.

HARTMANN, B.; LINK, N. Gesture recognition with inertial sensors and optimized dtw prototypes. In: **Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Istanbul, Turkey, 10-13 October 2010**. [S.l.]: IEEE, 2010. p. 2102–2109. Citado na página 77.

HASPERUE, W.; GIUSTI, L. A. D. Rule extraction on numeric datasets using hyper-rectangles. **Computer and Information Science**, v. 5, n. 4, p. 116–131, 2012. Citado na página 104.

HAYKIN, S. **Neural Networks: A Comprehensive Foundation**. New Jersey, USA: Prentice Hall, 1999. Citado na página 34.

HE, Q. et al. Fast time series classification based on infrequent shapelets. In: **11th IEEE International Conference on Machine Learning and Applications**. [S.l.: s.n.], 2012. Citado 3 vezes nas páginas 26, 27 e 77.

HERRERA, L. J. et al. Recursive prediction for long term time series forecasting using advanced models. **Neurocomput.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 70, n. 16-18, p. 2870–2880, out. 2007. ISSN 0925-2312. Citado na página 50.

HETLAND, M. L. A Survey of Recent Methods for Efficient Retrieval of Similar Time Sequences. In: LAST, M.; KANDEL, A.; BUNKE, H. (Ed.). **Data Mining in Time Series Databases**. [S.l.]: World Scientific, 2004, (Series in Machine Perception and Artificial Intelligence, v. 57). cap. 2, p. 27–49. ISBN 978-981-256-540-2. Citado na página 49.

HIDASI, B.; GASPAR-PAPANEK, C. Shifttree: An interpretable model-based approach for time series classification. In: GUNOPULOS, D. et al. (Ed.). **Machine Learning and Knowledge Discovery in Databases**. [S.l.]: Springer Berlin Heidelberg, 2011, (Lecture Notes in Computer Science, v. 6912). p. 48–64. ISBN 978-3-642-23782-9. Citado 3 vezes nas páginas 26, 57 e 103.

HILLS, J. et al. Classification of time series by shapelet transformation. **Data Mining and Knowledge Discovery**, Springer US, p. 1–31, 2013. ISSN 1384-5810. Citado 18 vezes nas páginas 26, 27, 29, 55, 59, 75, 76, 77, 79, 80, 81, 86, 87, 89, 92, 101, 103 e 144.

HOLTE, R. C.; ACKER, L. E.; PORTER, B. W. Concept learning and the problem of small disjuncts. In: **In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence**. [S.l.]: Morgan Kaufmann, 1989. p. 813–818. Citado na página 103.

HÖPPNER, F. Discovery of temporal patterns. learning rules about the qualitative behaviour of time series. In: **Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery**. London, UK, UK: Springer-Verlag, 2001. (PKDD '01), p. 192–203. ISBN 3-540-42534-9. Citado na página 50.



HUANG, Y.-W.; YU, P. S. Adaptive query processing for time-series data. In: **Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 1999. (KDD '99), p. 282–286. ISBN 1-58113-143-7. Citado na página [43](#).

HUGUENEY, B.; BOUCHON-MEUNIER, B. Time-series segmentation and symbolic representation, from process-monitoring to data-mining. In: **Proceedings of the International Conference, 7th Fuzzy Days on Computational Intelligence, Theory and Applications**. London, UK, UK: Springer-Verlag, 2001. p. 118–123. ISBN 3-540-42732-5. Citado na página [43](#).

IMAN, R. L.; DAVENPORT, J. M. Approximations of the critical region of the fbietkan statistic. **Communications in Statistics - Theory and Methods**, v. 9, n. 6, p. 571–595, 1980. Citado na página [88](#).

JANACEK, G.; BAGNALL, A.; POWELL, M. A likelihood ratio distance measure for the similarity between the fourier transform of time series. In: HO, T.; CHEUNG, D.; LIU, H. (Ed.). **Advances in Knowledge Discovery and Data Mining**. [S.l.]: Springer Berlin Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3518). p. 737–743. ISBN 978-3-540-26076-9. Citado 2 vezes nas páginas [47](#) e [49](#).

JEONG, Y.-S.; JEONG, M. K.; OMITAOMU, O. A. Weighted dynamic time warping for time series classification. **Pattern Recognition**, v. 44, n. 9, p. 2231–2240, 2011. ISSN 0031-3203. Computer Analysis of Images and Patterns. Citado 2 vezes nas páginas [45](#) e [53](#).

KADOUS, M. W. Learning comprehensible descriptions of multivariate time series. In: **ICML**. [S.l.: s.n.], 1999. p. 454–463. Citado 4 vezes nas páginas [26](#), [54](#), [56](#) e [103](#).

KADOUS, M. W.; SAMMUT, C. Constructive induction for classifying time series. In: **ECML**. [S.l.: s.n.], 2004. p. 192–204. Citado 4 vezes nas páginas [25](#), [26](#), [54](#) e [56](#).

KADOUS, M. W.; SAMMUT, C. Classification of multivariate time series and structured data using constructive induction. **Machine Learning**, v. 58, n. 2-3, p. 179–216, 2005. Citado 3 vezes nas páginas [26](#), [54](#) e [56](#).

KALPAKIS, K.; GADA, D.; PUTTAGUNTA, V. Distance measures for effective clustering of arima time-series. In: **Proceedings of the 2001 IEEE International Conference on Data Mining**. Washington, DC, USA: IEEE Computer Society, 2001. (ICDM '01), p. 273–280. ISBN 0-7695-1119-8. Citado na página [44](#).

KEOGH, E. et al. Dimensionality reduction for fast similarity search in large time series databases. **Knowledge and Information Systems**, Springer-Verlag London Limited, v. 3, n. 3, p. 263–286, 2001. ISSN 0219-1377. Citado na página [41](#).

KEOGH, E. et al. Locally adaptive dimensionality reduction for indexing large time series databases. **SIGMOD Rec.**, ACM, New York, NY, USA, v. 30, n. 2, p. 151–162, maio 2001. ISSN 0163-5808. Citado na página [42](#).

KEOGH, E.; KASETTY, S. On the need for time series data mining benchmarks: A survey and empirical demonstration. **Data Min. Knowl. Discov.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 7, n. 4, p. 349–371, out. 2003. ISSN 1384-5810. Citado na página [45](#).

KEOGH, E. et al. Finding the most unusual time series subsequence: Algorithms and applications. **Knowl. Inf. Syst.**, Springer-Verlag New York, Inc., New York, NY, USA, v. 11, n. 1, p. 1–27, dez. 2006. ISSN 0219-1377. Citado na página 50.

KEOGH, E.; LIN, J.; TRUPPEL, W. Clustering of time series subsequences is meaningless: implications for previous and future research. In: **Proceedings of the Third IEEE International Conference on Data Mining**. [S.l.]: IEEE Computer Society, 2003. p. 115–122. Citado 2 vezes nas páginas 37 e 61.

KEOGH, E.; LONARDI, S.; RATANAMAHATANA, C. A. Towards parameter-free data mining. In: **Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2004. (KDD '04), p. 206–215. ISBN 1-58113-888-1. Citado na página 48.

KEOGH, E. et al. Compression-based data mining of sequential data. **Data Min. Knowl. Discov.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 14, n. 1, p. 99–129, fev. 2007. ISSN 1384-5810. Citado na página 48.

KEOGH, E.; PAZZANI, M. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In: AGRAWAL, R.; STOLORZ, P.; PIATETSKY-SHAPIO, G. (Ed.). **Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)**. New York City, NY: ACM Press, 1998. p. 239–241. Citado 3 vezes nas páginas 25, 45 e 52.

KEOGH, E.; RATANAMAHATANA, C. A. Exact indexing of dynamic time warping. **Knowl. Inf. Syst.**, Springer-Verlag New York, Inc., New York, NY, USA, v. 7, n. 3, p. 358–386, mar. 2005. ISSN 0219-1377. Citado na página 45.

KEOGH, E. et al. **The UCR time series classification/clustering homepage**. 2006. Citado 4 vezes nas páginas 71, 85, 86 e 118.

KEOGH, E. J.; LIN, J. Clustering of time-series subsequences is meaningless: implications for previous and future research. **Knowl. Inf. Syst.**, v. 8, n. 2, p. 154–177, 2005. Citado na página 39.

KEOGH, E. J.; PAZZANI, M. J. Relevance feedback retrieval of time series data. In: **Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: ACM, 1999. (SIGIR '99), p. 183–190. ISBN 1-58113-096-1. Citado na página 38.

KEOGH, E. J.; PAZZANI, M. J. Derivative dynamic time warping. In: **In First SIAM International Conference on Data Mining (SDM'2001)**. [S.l.: s.n.], 2001. Citado 2 vezes nas páginas 45 e 53.

KEOGH, E. J.; RAKTHANMANON, T. Fast shapelets: A scalable algorithm for discovering time series shapelets. In: **SDM**. [S.l.: s.n.], 2013. p. 668–676. Citado 2 vezes nas páginas 71 e 77.

KOHAVI, R.; JOHN, G. H. Wrappers for feature subset selection. **Artif. Intell.**, Elsevier Science Publishers Ltd., Essex, UK, v. 97, n. 1-2, p. 273–324, dez. 1997. ISSN 0004-3702. Citado 2 vezes nas páginas 82 e 122.

KORN, F.; JAGADISH, H. V.; FALOUTSOS, C. Efficiently supporting ad hoc queries in large datasets of time sequences. In: **Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: ACM, 1997. (SIGMOD '97), p. 289–300. ISBN 0-89791-911-4. Citado 2 vezes nas páginas 41 e 42.

KOSKELA, T. **Neural Network Methods in Analysing and Modelling Time Varying Processes**. Tese (Doutorado) — Helsinki University of Technology, Espoo, Finland, Dezembro 2003. Citado na página 50.

KUDO, M.; TOYAMA, J.; SHIMBO, M. Multidimensional curve classification using passing-through regions. **Pattern Recognition Letters**, v. 20, n. 11–13, p. 1103 – 1111, 1999. ISSN 0167-8655. Citado 4 vezes nas páginas 26, 54, 56 e 103.

LAST, M.; KANDEL, A.; BUNKE, H. **Data Mining in Time Series Databases**. [S.l.]: World Scientific Publishing, 2004. Hardcover. ISBN 9812382909. Citado na página 25.

LAST, M.; KLEIN, Y.; KANDEL, A. Knowledge discovery in time series databases. **Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on**, v. 31, n. 1, p. 160–169, Feb 2001. ISSN 1083-4419. Citado 2 vezes nas páginas 25 e 50.

LAXMAN, S.; SASTRY, P. S. A survey of temporal data mining. **Sadhana**, Springer, v. 31, n. 2, p. 173–198, 2006. Citado 2 vezes nas páginas 25 e 37.

LEE, H. D. **Seleção de Atributos Importantes para a Extração de Conhecimento de Bases de Dados**. Tese (Tese de doutorado) — ICMC-USP, São Carlos, São Paulo, Brasil, 2005. Citado na página 82.

LIAO, T. W. Clustering of time series data—a survey. **Pattern Recognition**, v. 38, n. 11, p. 1857 – 1874, 2005. ISSN 0031-3203. Citado na página 49.

LIN, J. et al. A symbolic representation of time series, with implications for streaming algorithms. In: **Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery**. New York, NY, USA: ACM, 2003. (DMKD '03), p. 2–11. Citado na página 40.

LIN, J. et al. Experiencing sax: a novel symbolic representation of time series. **Data Min. Knowl. Discov.**, v. 15, n. 2, p. 107–144, 2007. Citado 2 vezes nas páginas 42 e 57.

LIN, J.; KHADE, R.; LI, Y. Rotation-invariant similarity in time series using bag-of-patterns representation. **Journal of Intelligent Information Systems**, Springer US, v. 39, n. 2, p. 287–315, 2012. ISSN 0925-9902. Citado na página 52.

LIN, J.; LI, Y. Finding structural similarity in time series data using bag-of-patterns representation. In: **Proceedings of the 21st International Conference on Scientific and Statistical Database Management**. Berlin, Heidelberg: Springer-Verlag, 2009. (SSDBM 2009), p. 461–477. ISBN 978-3-642-02278-4. Citado na página 52.

LIN, J. et al. A mpaa-based iterative clustering algorithm augmented by nearest neighbors search for time-series data streams. In: HO, T.; CHEUNG, D.; LIU, H. (Ed.). **Advances in Knowledge Discovery and Data Mining**. [S.l.]: Springer Berlin Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3518). p. 333–342. ISBN 978-3-540-26076-9. Citado na página 41.



LINES, J.; BAGNALL, A. Time series classification with ensembles of elastic distance measures. **Data Mining and Knowledge Discovery**, Springer US, v. 29, n. 3, p. 565–592, 2015. ISSN 1384-5810. Citado na página 84.

LINES, J. et al. A shapelet transform for time series classification. In: **KDD**. [S.l.: s.n.], 2012. p. 289–297. Citado 22 vezes nas páginas 26, 29, 42, 58, 59, 71, 72, 73, 74, 75, 76, 77, 79, 80, 81, 84, 87, 92, 101, 103, 143 e 147.

LIU, H.; MOTODA, H. **Feature Selection for Knowledge Discovery and Data Mining**. Norwell, MA, USA: Kluwer Academic Publishers, 1998. ISBN 079238198X. Citado na página 82.

LIU, H.; SETIONO, R. A probabilistic approach to feature selection - a filter solution. In: **13th International Conference on Machine Learning**. [S.l.: s.n.], 1996. p. 319–327. Citado na página 83.

LIU, Z. et al. Locating motifs in time-series data. In: **Proceedings of the 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining**. Berlin, Heidelberg: Springer-Verlag, 2005. (PAKDD'05), p. 343–353. ISBN 3-540-26076-5, 978-3-540-26076-9. Citado na página 51.

MALETZKE, A. G. **Uma metodologia para a extração de conhecimento em séries temporais por meio da identificação de *motifs* e da extração de características**. Dissertação (Mestrado) — Universidade de São Paulo - USP, 2009. Citado 4 vezes nas páginas 26, 36, 38 e 39.

MALETZKE, A. G. et al. Mineração de séries temporais por meio da extração de características e da identificação de motifs. In: **Congresso da Sociedade Brasileira de Computação**. [S.l.: s.n.], 2009. Citado 4 vezes nas páginas 28, 52, 57 e 58.

MALETZKE, A. G. et al. Time series classification with motifs and characteristics. In: ESPIN, R. et al. (Ed.). **Soft Computing for Business Intelligence**. [S.l.]: Springer Berlin Heidelberg, 2014, (Studies in Computational Intelligence, v. 537). p. 125–138. ISBN 978-3-642-53736-3. Citado 4 vezes nas páginas 25, 31, 57 e 103.

MANGANARIS, S. **Learning to classify sensor data**. Nashville, TN, USA, 1995. Citado na página 52.

MANNILA, H.; TOIVONEN, H.; VERKAMO, A. I. Discovery of frequent episodes in event sequences. **Data Min. Knowl. Discov.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 1, n. 3, p. 259–289, jan. 1997. ISSN 1384-5810. Citado na página 42.

MARTEAU, P.-F. Time warp edit distance with stiffness adjustment for time series matching. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 31, n. 2, p. 306–318, Feb 2009. ISSN 0162-8828. Citado na página 48.

MARTIN, B. **Instance-Based learning: Nearest Neighbor With Generalization**. Dissertação (Mestrado) — University of Waikato, Hamilton, New Zealand, 1995. Citado 3 vezes nas páginas 27, 29 e 104.

MCGOVERN, A. et al. Identifying predictive multi-dimensional time series motifs: an application to severe weather prediction. **Data Mining and Knowledge Discovery**, v. 22, n. 1, p. 232–258, 2011. Citado na página 77.

MEGALOOIKONOMOU, V.; LI, G.; WANG, Q. A dimensionality reduction technique for efficient similarity analysis of time series databases. In: **Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management**. New York, NY, USA: ACM, 2004. (CIKM '04), p. 160–161. ISBN 1-58113-874-1. Citado na página 43.

METZ, J. et al. Comparing published multi-label classifier performance measures to the ones obtained by a simple multi-label baseline classifier. **CoRR**, abs/1503.06952, 2015. Citado na página 85.

MICHALSKI, R. S.; BRATKO, I.; KUBAT, M. **Machine learning and data mining**. Chichester, West Sussex, England: Wiley, 1998. Citado 6 vezes nas páginas 26, 28, 31, 34, 38 e 55.

MITCHELL, T. M. **Machine Learning**. Boston, USA: McGraw-Hill, 1997. Citado 3 vezes nas páginas 31, 33 e 34.

MORCHEN, F. **Time series feature extraction for data mining using DWT and DFT**. Germany, 2003. Citado 2 vezes nas páginas 26 e 56.

MORCHEN, F. **Time series knowledge mining**. Tese (Tese de doutorado) — Department of Mathematics and Computer Science—Philipps-University, Marburg, Hesse, Germany, 2006. Citado 11 vezes nas páginas 25, 31, 37, 38, 40, 44, 47, 48, 49, 50 e 51.

MÖRCHEN, F.; ULTSCH, A. Optimizing time series discretization for knowledge discovery. In: **KDD**. [S.l.: s.n.], 2005. p. 660–665. Citado na página 42.

MORETTIN, P. A.; TOLOI, C. M. **Análise de Séries Temporais**. 2. ed. São Paulo, Brasil: Edgard Blecher, 2006. Citado 7 vezes nas páginas 25, 31, 35, 36, 37, 38 e 50.

MUEEN, A.; KEOGH, E.; YOUNG, N. Logical-shapelets: an expressive primitive for time series classification. In: **Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining**. New York, NY, USA: ACM, 2011. (KDD '11), p. 1154–1162. ISBN 978-1-4503-0813-7. Citado 8 vezes nas páginas 29, 58, 65, 70, 77, 84, 103 e 119.

MUEEN, A. et al. Exact discovery of time series motifs. In: **SDM**. [S.l.: s.n.], 2009. p. 473–484. Citado 2 vezes nas páginas 26 e 51.

NANOPOULOS, A.; ALCOCK, R.; MANOLOPOULOS, Y. Information processing and technology. In: MASTORAKIS, N.; NIKOLOPOULOS, S. D. (Ed.). Commack, NY, USA: Nova Science Publishers, Inc., 2001. cap. Feature-based Classification of Time-series Data, p. 49–61. ISBN 1-59033-116-8. Citado 3 vezes nas páginas 25, 44 e 52.

NEMENYI, P. **Distribution-free Multiple Comparisons**. [S.l.]: Princeton University, 1963. Citado na página 88.

PAGALLO, G.; HAUSSLER, D. Boolean feature discovery in empirical learning. **Machine Learning**, Kluwer Academic Publishers-Plenum Publishers, v. 5, n. 1, p. 71–99, 1990. ISSN 0885-6125. Citado na página 103.

PANUCCIO, A.; BICEGO, M.; MURINO, V. A hidden markov model-based approach to sequential data clustering. In: **Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition**. London, UK, UK: Springer-Verlag, 2002. p. 734–742. ISBN 3-540-44011-9. Citado 2 vezes nas páginas 44 e 48.

PATEL, P. et al. Mining motifs in massive time series databases. In: **Proceedings of the 2002 IEEE International Conference on Data Mining**. Washington, DC, USA: IEEE Computer Society, 2002. (ICDM '02), p. 370–. ISBN 0-7695-1754-4. Citado na página 51.

POPIVANOV, I.; MILLER, R. Similarity search over time-series data using wavelets. In: **Data Engineering, 2002. Proceedings. 18th International Conference on**. [S.l.: s.n.], 2002. p. 212–221. ISSN 1063-6382. Citado na página 41.

PRESS, W. H. et al. **Numerical Recipes in C (2Nd Ed.): The Art of Scientific Computing**. New York, NY, USA: Cambridge University Press, 1992. ISBN 0-521-43108-5. Citado na página 83.

PYLE, D. **Data Preparation for Data Mining**. Califórnia, USA: Morgan Kaufmann, 1999. Citado 3 vezes nas páginas 32, 35 e 38.

QU, Y.; WANG, C.; WANG, X. S. Supporting fast search in time series for movement patterns in multiple scales. In: **Proceedings of the Seventh International Conference on Information and Knowledge Management**. New York, NY, USA: ACM, 1998. (CIKM '98), p. 251–258. ISBN 1-58113-061-9. Citado na página 43.

QUINLAN, J. R. Induction of decision trees. In: \_\_\_\_\_. **Readings in Machine Learning**. [S.l.]: Morgan Kaufmann, 1986. v. 1, cap. Readings in Machine Learning, p. 81–106. Citado na página 103.

QUINLAN, J. R. **C4.5: Programs for Machine Learning**. [S.l.]: Morgan Kaufmann, 1993. ISBN 1-55860-238-0. Citado 2 vezes nas páginas 33 e 103.

RAKTHANMANON, T.; KEOGH, E. Fast shapelets: A scalable algorithm for discovering time series shapelets. In: **SDM**. [S.l.: s.n.], 2013. Citado 9 vezes nas páginas 26, 27, 28, 84, 86, 94, 103, 119 e 134.

RAKTHANMANON, T. et al. Time series epenthesis: Clustering time series streams requires ignoring some data. In: **ICDM**. [S.l.: s.n.], 2011. p. 547–556. Citado na página 28.

RANI, S.; SIKKA, G. Article: Recent techniques of clustering of time series data: A survey. **International Journal of Computer Applications**, v. 52, n. 15, p. 1–9, August 2012. Citado na página 49.

RATANAMAHATANA, C. et al. A novel bit level time series representation with implication of similarity search and clustering. In: **Proceedings of the 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining**. Berlin, Heidelberg: Springer-Verlag, 2005. (PAKDD'05), p. 771–777. ISBN 3-540-26076-5, 978-3-540-26076-9. Citado na página 43.

REISS, A.; WEBER, M.; STRICKER, D. Exploring and extending the boundaries of physical activity recognition. In: **IEEE International Conference on Systems, Man, and Cybernetics**. [S.l.: s.n.], 2011. Citado na página [77](#).

REZENDE, S. O. **Sistemas Inteligentes: Fundamentos e Aplicações**. Barueri, Brasil: Manole, 2003. Citado 5 vezes nas páginas [26](#), [28](#), [32](#), [33](#) e [55](#).

RODRIGUEZ, J. J.; ALONSO, C. J.; BOSTROM, H. Boosting interval based literals. **Intell. Data Anal.**, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 5, n. 3, p. 245–262, ago. 2001. ISSN 1088-467X. Citado 3 vezes nas páginas [26](#), [54](#) e [56](#).

RODRÍGUEZ, J. J.; ALONSO, C. J.; MAESTRO, J. A. Support vector machines of interval-based features for time series classification. **Knowledge-Based Systems**, v. 18, n. 4–5, p. 171 – 178, 2005. ISSN 0950-7051. AI-2004, Cambridge, England, 13th-15th December 2004. Citado na página [52](#).

ROVERSO, D. Multivariate temporal classification by windowed wavelet decomposition and recurrent neural networks. In: **In 3rd ANS International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface**. [S.l.: s.n.], 2000. Citado na página [52](#).

RUDELL, R. Dynamic variable ordering for ordered binary decision diagrams. In: **Computer-Aided Design, 1993. ICCAD-93. Digest of Technical Papers., 1993 IEEE/ACM International Conference on**. [S.l.: s.n.], 1993. p. 42–47. Citado na página [111](#).

SAKOE, H.; CHIBA, S. Dynamic programming algorithm optimization for spoken word recognition. **Acoustics, Speech and Signal Processing, IEEE Transactions on**, v. 26, n. 1, p. 43–49, Feb 1978. ISSN 0096-3518. Citado na página [45](#).

SALVADOR, S.; CHAN, P. Toward accurate dynamic time warping in linear time and space. **Intell. Data Anal.**, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 11, n. 5, p. 561–580, out. 2007. ISSN 1088-467X. Citado na página [45](#).

SALZBERG, S. A nearest hyperrectangle learning method. **Mach. Learn.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 6, n. 3, p. 251–276, maio 1991. ISSN 0885-6125. Citado 3 vezes nas páginas [29](#), [104](#) e [145](#).

SEBASTIANI, P.; RAMONI, M. Clustering continuous time series. In: **Proceedings of the Eighteenth International Conference on Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001. (ICML '01), p. 497–504. ISBN 1-55860-778-1. Citado na página [44](#).

SENIN, P.; MALINCHIK, S. Sax-vsm: Interpretable time series classification using sax and vector space model. In: **Data Mining (ICDM), 2013 IEEE 13th International Conference on**. [S.l.: s.n.], 2013. p. 1175–1180. ISSN 1550-4786. Citado na página [52](#).

SERRA, J.; ARCOS, J. L. An empirical evaluation of similarity measures for time series classification. **Knowledge-Based Systems**, v. 67, n. 0, p. 305–314, 2014. ISSN 0950-7051. Citado na página [53](#).

SFETSOS, A.; SIRIOPOULOS, C. Time series forecasting with a hybrid clustering scheme and pattern recognition. **Trans. Sys. Man Cyber. Part A**, IEEE Press, Piscataway, NJ, USA, v. 34, n. 3, p. 399–405, maio 2004. ISSN 1083-4427. Citado na página [50](#).

SHANNON, C. E. A mathematical theory of communication. **Bell system technical journal**, v. 27, 1948. Citado na página [66](#).

SHASHA, D.; ZHU, Y. **High Performance Discovery In Time Series: Techniques And Case Studies (Monographs in Computer Science)**. [S.l.]: SpringerVerlag, 2004. ISBN 0387008578. Citado na página [41](#).

SHATKAY, H.; ZDONIK, S. B. Approximate queries and representations for large data sequences. In: **Proceedings of the Twelfth International Conference on Data Engineering**. Washington, DC, USA: IEEE Computer Society, 1996. (ICDE '96), p. 536–545. ISBN 0-8186-7240-4. Citado na página [42](#).

SHUMWAY, R. H.; STOFFER, D. S. **Time Series Analysis and its Applications: with R examples**. 2. ed. New York, USA: Springer, 2006. Citado na página [38](#).

SILVA, D. F.; SOUZA, V. M. A. de; BATISTA, G. E. Time series classification using compression distance of recurrence plots. In: . [S.l.: s.n.], 2013. p. 687–696. Citado 5 vezes nas páginas [44](#), [47](#), [53](#), [94](#) e [134](#).

SILVA, L. M. O. da. **Uma Aplicação de árvores de decisão, redes neurais e KNN para a identificação de modelos ARMA não-Sazonais e sazonais**. Tese (Doutorado) — Pontificada Universidade Católica do Rio de Janeiro - PUC-RJ, Rio de Janeiro, RJ, Setembro 2005. Citado na página [36](#).

SIVAKUMAR, P.; SHAJINA, T. Human gait recognition and classification using time series shapelets. In: **IEEE International Conference on Advances in Computing and Communications**. [S.l.: s.n.], 2012. Citado na página [77](#).

SOMENZI, F. Binary decision diagrams. In: **Calculational System Design, volume 173 of NATO Science Series F: Computer and Systems Sciences**. [S.l.]: IOS Press, 1999. p. 303–366. Citado na página [112](#).

SOUZA, V.; SILVA, D.; BATISTA, G. Extracting texture features for time series classification. In: **Pattern Recognition (ICPR), 2014 22nd International Conference on**. [S.l.: s.n.], 2014. p. 1425–1430. ISSN 1051-4651. Citado 2 vezes nas páginas [25](#) e [53](#).

SRIKANT, R.; AGRAWAL, R. Mining sequential patterns: Generalizations and performance improvements. In: **Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology**. London, UK, UK: Springer-Verlag, 1996. (EDBT '96), p. 3–17. ISBN 3-540-61057-X. Citado na página [49](#).

STEFAN, A.; ATHITSOS, V.; DAS, G. The move-split-merge metric for time series. **Knowledge and Data Engineering, IEEE Transactions on**, v. 25, n. 6, p. 1425–1438, June 2013. ISSN 1041-4347. Citado na página [53](#).



VLACHOS, M.; GUNOPOULOS, D.; KOLLIOS, G. Discovering similar multidimensional trajectories. In: **Proceedings of the 18th International Conference on Data Engineering**. Washington, DC, USA: IEEE Computer Society, 2002. (ICDE '02), p. 673–684. Citado na página [48](#).

VLACHOS, M. et al. Indexing multi-dimensional time-series with support for multiple distance measures. In: **Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2003. (KDD '03), p. 216–225. ISBN 1-58113-737-0. Citado na página [49](#).

VLACHOS, M.; YU, P. S.; CASTELLI, V. On periodicity detection and structural periodic similarity. In: KARGUPTA, H. et al. (Ed.). **SDM**. [S.l.]: SIAM, 2005. p. 449–460. ISBN 978-1-61197-275-7. Citado na página [47](#).

WANG, H.; SONG, M. Ckmeans.1d.dp: Optimal  $k$ -means clustering in one dimension by dynamic programming. **The R Journal**, v. 3, n. 2, p. 29–33, 2011. Citado na página [107](#).

WANG, J.; HAN, J. Bide: Efficient mining of frequent closed sequences. In: **Proceedings of the 20th International Conference on Data Engineering**. Washington, DC, USA: IEEE Computer Society, 2004. (ICDE '04), p. 79–90. ISBN 0-7695-2065-0. Citado 2 vezes nas páginas [44](#) e [52](#).

WANG, X. et al. Experimental comparison of representation methods and distance measures for time series data. **Data Min. Knowl. Discov.**, v. 26, n. 2, p. 275–309, 2013. Citado 6 vezes nas páginas [25](#), [53](#), [55](#), [84](#), [103](#) e [122](#).

WEISS, G. M. Mining with rarity: A unifying framework. **SIGKDD Explor. Newsl.**, ACM, New York, NY, USA, v. 6, n. 1, p. 7–19, jun. 2004. ISSN 1931-0145. Citado na página [50](#).

WEISS, S. M.; INDURKHYA, N. **Predictive Data Mining: a practical guide**. California, USA: Morgan Kaufmann, 1998. Citado 2 vezes nas páginas [31](#) e [32](#).

WETTSCHERECK, D. A hybrid nearest-neighbor and nearest-hyperrectangle algorithm. In: BERGADANO, F.; RAEDT, L. D. (Ed.). **Machine Learning: ECML-94**. [S.l.]: Springer Berlin Heidelberg, 1994, (Lecture Notes in Computer Science, v. 784). p. 323–335. ISBN 978-3-540-57868-0. Citado na página [104](#).

WISTUBA, M.; GRABOCKA, J.; SCHMIDT-THIEME, L. Ultra-Fast Shapelets for Time Series Classification. **ArXiv e-prints**, mar. 2015. Citado na página [77](#).

WITTEN, I. H.; FRANK, E. **Data mining: practical machine learning tools and techniques**. 2. ed. San Francisco, California, USA: Elsevier, 2005. Citado na página [32](#).

XI, X. et al. Fast time series classification using numerosity reduction. In: **ICML**. [S.l.: s.n.], 2006. p. 1033–1040. Citado na página [55](#).

XING, Z.; PEI, J.; KEOGH, E. J. A brief survey on sequence classification. **SIGKDD Explorations**, v. 12, n. 1, p. 40–48, 2010. Citado na página [25](#).

XING, Z.; PEI, J.; YU, P. Early classification on time series. **Knowledge and information systems**, v. 31, n. 1, p. 105–127, 2012. Citado 4 vezes nas páginas 26, 53, 55 e 76.

XING, Z. et al. Extracting interpretable features for early classification on time series. In: **SDM**. [S.l.: s.n.], 2011. p. 247–258. Citado 5 vezes nas páginas 26, 53, 55, 76 e 147.

XIONG, Y.; YEUNG, D.-Y. Time series clustering with arma mixtures. **Pattern Recognition**, v. 37, n. 8, p. 1675 – 1689, 2004. ISSN 0031-3203. Citado na página 48.

YADAV, R. N.; KALRA, P. K.; JOHN, J. Time series prediction with single multiplicative neuron model. **Appl. Soft Comput.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 7, n. 4, p. 1157–1163, ago. 2007. ISSN 1568-4946. Citado na página 50.

YAMADA, Y. et al. Decision-tree induction from time-series data based on a standard-example split test. In: **ICML**. [S.l.: s.n.], 2003. p. 840–847. Citado 8 vezes nas páginas 28, 53, 55, 56, 58, 59, 60 e 103.

YAMADA, Y. et al. Experimental evaluation of time-series decision tree. In: TSUMOTO, S. et al. (Ed.). **Active Mining**. [S.l.]: Springer Berlin Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3430). p. 190–209. ISBN 978-3-540-26157-5. Citado 2 vezes nas páginas 26 e 53.

YANKOV, D. et al. Detecting time series motifs under uniform scaling. In: **Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2007. (KDD '07), p. 844–853. ISBN 978-1-59593-609-7. Citado na página 51.

YE, L.; KEOGH, E. Time series shapelets: a new primitive for data mining. In: **Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining**. New York, NY, USA: ACM, 2009. (KDD '09), p. 947–956. ISBN 978-1-60558-495-9. Citado 18 vezes nas páginas 26, 28, 42, 51, 53, 54, 58, 60, 63, 67, 70, 72, 73, 75, 76, 77, 79 e 103.

YE, L.; KEOGH, E. J. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. **Data Min. Knowl. Discov.**, v. 22, n. 1-2, p. 149–182, 2011. Citado 10 vezes nas páginas 26, 28, 53, 55, 58, 63, 64, 68, 70 e 103.

YI, B.-K.; FALOUTSOS, C. Fast time sequence indexing for arbitrary lp norms. In: **Proceedings of the 26th International Conference on Very Large Data Bases**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000. (VLDB '00), p. 385–394. ISBN 1-55860-715-3. Citado na página 45.

YU, L.; LIU, H. Feature selection for high-dimensional data: A fast correlation-based filter solution. In: **Proceedings of the Twentieth International Conference on Machine Learning**. [S.l.]: AAAI Press, 2003. p. 856–863. Citado na página 83.

YUAN, J.-D.; WANG, Z.-H.; HAN, M. A discriminative shapelets transformation for time series classification. **International Journal of Pattern Recognition and Artificial Intelligence**, v. 28, n. 06, p. 1450014, 2014. Citado 4 vezes nas páginas 26, 76, 77 e 103.



ZAKARIA, J.; MUEEN, A.; KEOGH, E. Clustering time series using unsupervised-shapelets. In: **Proceedings of the 2012 IEEE 12th International Conference on Data Mining**. Washington, DC, USA: IEEE Computer Society, 2012. (ICDM '12), p. 785–794. ISBN 978-0-7695-4905-7. Citado 2 vezes nas páginas [28](#) e [77](#).

ZALEWSKI, W. et al. Time series discretization based on the approximation of the local slope information. In: **Advances in Artificial Intelligence - IBERAMIA 2012 - 13th Ibero-American Conference on AI, Cartagena de Indias, Colombia, November 13-16, 2012. Proceedings**. [S.l.: s.n.], 2012b. p. 91–100. Citado 2 vezes nas páginas [43](#) e [57](#).

ZALEWSKI, W. et al. Symbolic representation based on temporal order information for time series classification. In: **Brazilian Conference on Intelligent Systems, BRACIS 2013, Fortaleza, CE, Brazil, 19-24 October, 2013**. [S.l.: s.n.], 2013. p. 95–100. Citado 2 vezes nas páginas [43](#) e [57](#).

ZALEWSKI, W. et al. A symbolic representation method to preserve the characteristic slope of time series. In: **Advances in Artificial Intelligence - SBIA 2012 - 21th Brazilian Symposium on Artificial Intelligence, Curitiba, Brazil, October 20-25, 2012. Proceedings**. [S.l.: s.n.], 2012a. p. 132–141. Citado na página [43](#).

ZHANG, D. et al. Time series classification using support vector machine with gaussian elastic metric kernel. In: **Pattern Recognition (ICPR), 2010 20th International Conference on**. [S.l.: s.n.], 2010. p. 29–32. ISSN 1051-4651. Citado na página [53](#).

ZHANG, H.; HO, T.; LIN, M. A non-parametric wavelet feature extractor for time series classification. In: DAI, H.; SRIKANT, R.; ZHANG, C. (Ed.). **Advances in Knowledge Discovery and Data Mining**. [S.l.]: Springer Berlin Heidelberg, 2004, (Lecture Notes in Computer Science, v. 3056). p. 595–603. ISBN 978-3-540-22064-0. Citado na página [52](#).

ZHONG, S.; GHOSH, J. Hmms and coupled hmms for multi-channel eeg classification. In: **Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on**. [S.l.: s.n.], 2002. v. 2, p. 1154–1159. ISSN 1098-7576. Citado na página [52](#).

## APÊNDICES

## APÊNDICE A – QUANTIDADE DE ATRIBUTOS (*SHAPELETS*)

Neste apêndice são apresentados os resultados referentes à quantidade de atributos utilizada pelos modelos construídos por meio dos algoritmos J48, HID-*wrapper*, NNGE-*wrapper* e SSA (NNGE e HID), para cada uma das estratégias de aprendizado de limiares propostas no capítulo 5.

Tabela A.1 – Quantidade de atributos utilizada pelos modelos construídos utilizando os algoritmos J48, HID (usando *wrapper*), NNGE (usando *wrapper*) e SSA (NNGE e HID), considerando a estratégia de aprendizado de limiares, ganho de informação (lg). A última linha da tabela representa a soma da quantidade de atributos de todos os conjuntos de dados avaliados.

#	lg-TS+CFS				lg-TS+CS				lg-TS+FCBF			
	J48	SSA	HID	NNGE	J48	SSA	HID	NNGE	J48	SSA	HID	NNGE
1	18	27	20	20	20	142	35	31	30	233	33	41
2	16	27	19	22	23	100	35	39	39	191	63	56
3	4	40	4	3	5	43	9	3	5	79	5	4
4	2	3	2	2	2	14	2	2	2	8	2	2
5	5	281	10	10	9	29	13	9	4	36	7	10
6	1	133	1	1	1	34	1	1	1	133	1	1
7	29	190	49	48	30	138	34	17	17	203	58	35
8	30	130	45	52	19	117	24	24	33	169	35	24
9	24	201	39	50	28	134	17	25	36	192	47	30
10	2	21	2	3	2	33	2	3	2	26	2	3
11	3	28	3	4	3	23	3	9	2	37	5	5
12	1	39	1	1	1	16	1	1	1	38	1	1
13	26	36	23	23	24	153	39	28	22	172	42	25
14	2	9	2	3	2	22	2	3	3	42	3	3
15	11	31	18	17	13	99	29	15	9	145	21	19
16	6	16	6	7	7	85	11	10	7	170	11	9
17	1	25	1	1	1	20	3	2	1	26	1	1
18	1	10	1	3	1	10	4	4	1	15	3	1
19	1	65	4	10	3	59	4	4	2	135	3	3
20	4	20	7	5	5	41	14	6	7	107	11	12
21	11	23	17	16	14	50	24	18	15	49	16	9
22	1	45	1	1	1	19	3	2	1	44	1	1
23	2	7	2	3	4	20	3	3	3	19	3	3
24	8	29	12	17	11	91	21	21	9	167	11	15
25	1	42	1	1	1	11	1	1	1	48	1	1
26	12	32	19	20	16	119	25	24	19	200	26	22
27	4	16	4	4	6	12	6	5	5	7	5	5
28	5	18	6	4	10	60	12	16	8	54	11	13
29	2	8	2	3	2	15	2	3	3	40	3	5
30	1	58	1	1	1	15	1	1	1	54	1	1
31	28	195	53	90	9	35	9	7	10	29	8	13
32	13	36	16	26	18	127	34	26	25	209	48	31
Total	275	1841	391	471	292	1886	423	363	324	3077	488	404

Tabela A.2 – Quantidade de atributos utilizada pelos modelos construídos utilizando os algoritmos J48, HID (usando *wrapper*), NNGE (usando *wrapper*) e SSA (NNGE e HID), considerando a estratégia de aprendizado de limiares, agrupamento (2k). A última linha da tabela representa a soma da quantidade de atributos de todos os conjuntos de dados avaliados.

#	2k-TS+CFS				2k-TS+CS				2k-TS+FCBF			
	J48	SSA	HID	NNGE	J48	SSA	HID	NNGE	J48	SSA	HID	NNGE
1	24	127	33	40	11	16	11	13	10	19	9	18
2	18	20	19	20	14	19	18	19	15	18	13	18
3	3	3	3	3	5	5	5	5	3	4	4	4
4	2	2	2	2	2	2	2	2	2	2	2	2
5	7	16	11	12	8	10	8	9	6	10	8	9
6	1	1	1	1	1	1	1	1	1	1	1	1
7	30	145	54	64	13	15	12	15	12	13	12	13
8	35	104	50	62	7	14	10	11	12	15	15	14
9	26	156	44	48	10	14	9	13	9	14	11	11
10	2	2	2	2	2	2	2	2	2	3	2	3
11	3	4	3	4	3	9	3	3	2	5	3	4
12	1	1	1	1	1	1	1	1	1	1	1	1
13	30	53	36	48	11	13	13	13	12	14	13	14
14	2	2	2	2	2	3	2	2	3	3	3	3
15	12	28	17	21	8	10	9	9	7	10	8	10
16	5	6	6	5	8	8	7	7	5	6	6	6
17	1	1	1	1	1	2	1	2	1	1	1	1
18	1	2	1	2	1	3	1	2	1	2	1	1
19	1	3	2	3	3	4	4	4	2	3	2	3
20	4	5	5	5	4	6	5	6	5	6	6	6
21	14	43	25	32	17	21	13	17	10	17	13	14
22	1	1	1	1	1	2	1	2	1	1	1	1
23	2	2	2	2	3	3	3	3	3	3	3	3
24	8	13	10	12	9	10	10	8	6	8	7	7
25	1	1	1	1	1	1	1	1	1	1	1	1
26	14	21	15	19	10	13	11	12	11	13	13	13
27	3	3	3	3	4	5	4	5	5	5	5	5
28	8	10	8	8	7	10	8	10	7	9	7	9
29	2	2	2	2	2	2	2	2	2	3	2	2
30	1	1	1	1	1	1	1	1	1	1	1	1
31	10	28	13	23	13	16	12	15	16	28	15	28
32	16	79	27	44	10	13	8	12	9	13	8	11
Total	288	885	401	494	193	254	198	227	183	252	197	237

Tabela A.3 – Quantidade de atributos utilizada pelos modelos construídos utilizando os algoritmos J48, HID (usando *wrapper*), NNGE (usando *wrapper*) e SSA (NNGE e HID), considerando a estratégia de aprendizado de limiares, precisão (Pr). A última linha da tabela representa a soma da quantidade de atributos de todos os conjuntos de dados avaliados.

#	Pr-TS+CFS				Pr-TS+CS				Pr-TS+FCBF			
	J48	SSA	HID	NNGE	J48	SSA	HID	NNGE	J48	SSA	HID	NNGE
1	16	126	28	35	25	121	16	17	18	317	40	37
2	14	37	18	23	16	68	16	28	42	155	32	32
3	5	16	4	3	6	36	9	9	4	53	6	5
4	2	3	2	2	2	14	2	2	2	8	2	2
5	5	279	12	12	8	27	10	11	4	35	8	8
6	1	1	1	1	1	1	1	1	1	1	1	1
7	26	202	50	61	20	132	26	17	22	199	19	44
8	29	129	43	49	15	105	28	21	20	149	39	21
9	24	197	42	46	13	134	19	32	18	201	29	24
10	2	21	2	3	2	11	2	3	2	28	2	3
11	3	23	3	4	3	21	3	3	2	30	5	6
12	1	1	1	1	1	1	1	1	1	1	1	1
13	25	94	35	39	19	135	36	16	19	139	40	22
14	2	9	2	3	2	14	2	3	3	19	3	3
15	9	33	13	20	16	80	26	17	10	130	30	17
16	5	12	6	5	7	59	11	11	7	112	9	11
17	1	1	1	1	1	16	3	2	1	1	1	1
18	1	6	3	3	1	9	4	4	1	10	3	1
19	1	40	4	9	2	55	4	4	2	128	3	3
20	4	17	7	8	5	36	7	8	8	85	11	14
21	11	23	17	16	9	27	12	12	9	31	15	16
22	1	1	1	1	1	11	3	2	1	1	1	1
23	2	7	2	3	4	16	3	3	3	25	3	3
24	7	24	8	12	12	76	30	10	6	103	13	11
25	1	1	1	1	1	1	1	1	1	1	1	1
26	12	18	11	12	20	96	15	20	24	115	23	20
27	3	12	3	4	6	10	6	5	7	9	5	5
28	3	13	3	3	7	18	10	13	9	15	11	10
29	2	6	2	4	2	13	2	3	2	15	4	3
30	1	1	1	1	1	1	1	1	1	1	1	1
31	26	196	49	87	9	40	11	18	6	26	8	13
32	17	62	22	36	15	112	19	26	17	272	36	18
Total	262	1611	397	508	252	1496	339	324	273	2415	405	358

## APÊNDICE B – RESULTADOS DE ACURÁCIA PARA SSA (HID E NNGE)

Neste apêndice são apresentados os resultados dos experimentos realizados no capítulo 5 utilizando os algoritmos HID e NNGE, ambos sem seleção de atributos (SSA), considerando cada uma das estratégias de binarização propostas.

Tabela B.1 – Resultados dos experimentos realizados utilizando o HID como algoritmo base, sem seleção de atributos (SSA). Os valores são apresentados no formato *acc (rank)*, no qual *acc* corresponde ao valor da acurácia (%) de classificação sobre o conjunto de teste.

#	TS+CFS			TS+CS			TS+FCBF		
	lg	2k	Pr	lg	2k	Pr	lg	2k	Pr
1	39,56	65,27	46,81	38,46	50,11	32,31	12,97	37,80	24,62
2	82,10	81,33	84,40	88,49	71,36	50,38	58,57	44,25	56,01
3	83,33	53,33	93,33	66,67	56,67	53,33	80,00	50,00	83,33
4	83,00	93,22	88,56	85,11	91,78	91,11	81,89	88,56	86,22
5	90,57	93,70	95,26	94,30	95,55	92,14	88,78	90,81	95,21
6	96,43	92,86	96,43	96,43	100,00	96,43	96,43	92,86	96,43
7	38,72	77,44	75,64	42,31	53,85	58,46	32,31	48,97	56,92
8	47,44	80,26	82,56	49,23	50,51	64,62	41,54	46,92	53,08
9	42,05	76,41	74,62	45,13	54,87	56,41	39,23	49,74	53,59
10	79,09	76,80	80,72	77,78	70,59	97,06	69,93	61,44	82,68
11	84,00	82,00	82,00	83,00	76,00	86,00	85,00	83,00	82,00
12	99,30	97,79	99,30	98,49	97,33	98,49	98,49	96,86	98,49
13	68,05	79,35	77,93	70,59	65,44	68,64	60,00	74,44	75,33
14	55,68	84,09	81,82	56,82	82,95	85,23	72,73	87,50	87,50
15	76,49	88,98	86,20	79,41	67,51	69,51	57,80	79,46	80,54
16	90,86	98,29	99,43	91,43	94,29	89,14	81,71	94,86	95,43
17	99,33	100,00	99,33	92,67	97,33	92,67	92,67	100,00	92,67
18	96,79	97,08	96,50	91,16	96,40	94,46	95,43	96,40	96,31
19	54,10	60,66	77,05	68,85	45,90	55,74	55,74	54,10	63,93
20	64,38	67,12	67,12	68,49	71,23	61,64	67,12	50,68	57,53
21	68,68	79,21	80,79	75,26	74,08	77,63	68,16	63,95	70,13
22	93,69	88,50	94,01	85,14	80,99	85,14	85,14	87,94	85,14
23	93,33	96,67	96,67	86,67	96,67	100,00	90,00	96,67	83,33
24	78,51	94,63	97,52	75,21	77,69	75,62	78,10	86,78	90,50
25	92,51	78,37	93,01	86,02	53,24	86,02	86,02	86,19	86,02
26	90,56	94,88	94,72	88,16	88,32	87,68	84,00	83,04	92,48
27	74,07	60,10	66,43	80,00	66,53	69,45	76,28	52,46	68,64
28	84,33	91,00	95,67	94,33	93,33	94,33	81,33	81,67	88,00
29	100,00	99,00	100,00	100,00	100,00	99,00	100,00	98,00	99,00
30	100,00	100,00	100,00	96,75	94,56	96,75	96,75	94,56	96,75
31	72,80	76,08	83,38	73,53	86,53	86,33	72,65	73,33	83,43
32	47,81	65,05	57,68	48,28	53,29	42,79	43,26	33,07	37,46
acc média	77,11	83,42	85,78	77,32	76,72	77,95	72,81	73,95	78,08

Tabela B.2 – Resultados dos experimentos realizados utilizando o NNGE como algoritmo base, sem seleção de atributos (SSA). Os valores são apresentados no formato *acc (rank)*, no qual *acc* corresponde ao valor da acurácia (%) de classificação sobre o conjunto de teste.

#	TS+CFS			TS+CS			TS+FCBF		
	lg	2k	Pr	lg	2k	Pr	lg	2k	Pr
1	32,09	68,35	51,21	37,80	53,85	38,24	11,21	52,53	42,64
2	80,31	90,03	82,61	90,03	78,77	50,90	59,34	71,36	68,03
3	83,33	53,33	86,67	63,33	46,67	50,00	86,67	66,67	80,00
4	85,11	95,89	84,78	85,11	91,78	91,11	85,11	95,89	83,56
5	90,57	95,78	93,72	94,71	95,49	92,34	88,78	95,10	94,40
6	96,43	92,86	96,43	96,43	100,00	96,43	96,43	92,86	96,43
7	41,54	63,08	54,36	47,18	52,05	52,82	30,26	58,97	51,28
8	46,67	69,74	71,03	50,00	53,59	62,82	42,82	54,87	61,03
9	42,56	64,10	56,67	42,56	56,41	56,41	33,59	53,08	53,85
10	85,29	80,72	82,68	77,78	70,59	97,06	73,86	72,55	85,29
11	86,00	79,00	80,00	83,00	76,00	86,00	84,00	74,00	84,00
12	99,07	97,21	99,07	98,49	97,33	98,49	98,49	96,86	98,49
13	58,76	71,24	69,76	71,18	68,40	70,30	50,83	65,21	64,44
14	68,18	90,91	89,77	56,82	82,95	90,91	76,14	86,36	86,36
15	70,39	82,59	74,05	77,46	66,68	67,85	57,90	77,51	70,24
16	92,57	93,14	95,43	91,43	92,57	89,71	84,57	97,14	94,86
17	98,67	95,33	96,67	92,67	97,33	92,67	92,67	96,67	92,67
18	96,40	96,60	96,21	91,16	96,40	94,46	94,85	96,70	92,52
19	55,74	60,66	62,30	68,85	47,54	55,74	63,93	54,10	55,74
20	71,23	61,64	58,90	64,38	65,75	58,90	64,38	58,90	49,32
21	67,89	78,68	82,37	75,53	73,95	80,79	67,37	68,95	79,47
22	93,21	89,70	93,53	85,14	80,99	85,14	85,14	85,78	85,14
23	93,33	100,00	93,33	86,67	96,67	100,00	93,33	100,00	93,33
24	70,25	81,82	79,34	73,55	78,51	70,25	76,86	73,97	83,88
25	92,85	76,54	92,68	86,02	53,24	86,02	86,02	86,19	86,02
26	84,64	93,92	90,88	85,44	86,08	85,76	70,08	86,08	84,64
27	76,88	70,45	68,14	83,12	66,53	69,45	79,60	67,04	70,05
28	88,33	93,00	94,00	95,33	93,33	94,33	82,33	86,67	89,33
29	100,00	100,00	100,00	100,00	100,00	99,00	100,00	98,00	100,00
30	100,00	98,42	100,00	96,75	94,56	96,75	96,75	94,56	96,75
31	73,13	82,13	77,88	76,95	85,85	86,68	73,15	83,50	78,48
32	47,49	63,17	48,59	46,24	51,41	41,38	42,16	43,42	41,54
acc média	77,15	82,19	81,35	77,22	76,60	77,77	72,77	77,86	77,93